EP0884912

Publication Title:

Image sequence coding and decoding method

Abstract:

In order to prevent accumulation of rounding errors caused by the integer-based bilinear interpolation used in the motion compensation process of image coding and decoding methods, two types of P frames are used for unidirectional motion compensation prediction, namely:

P+ frames, which round the results of bilinear interpolation with real number operations to the nearest integer and rounds half integer values (0.5 added to an integer) away from zero, and

P- frames, which differs from p+ frames in that the above mentioned half integer values are rounded towards zero.

Utilising both of these P+ and P- frames enables cancelling of rounding errors and prevents the accumulation of rounding errors.

Data supplied from the esp@cenet database - http://ep.espacenet.com

This Patent PDF Generated by Patent Fetcher(TM), a service of Patent Logistics, LLC

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 884 912 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
16.12.1998 Bulletin 1998/51

(51) Int. Cl.[6]: **H04N 7/36**

(21) Application number: **98110471.4**

(22) Date of filing: **08.06.1998**

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: **09.06.1997 JP 150656/97**

(71) Applicant: **Hitachi, Ltd.**
Chiyoda-ku, Tokyo 101-8010 (JP)

(72) Inventor: **Nakaya, Yuichiro**
Tokyo, 167-0041 (JP)

(74) Representative:
Strehl Schübel-Hopf & Partner
Maximilianstrasse 54
80538 München (DE)

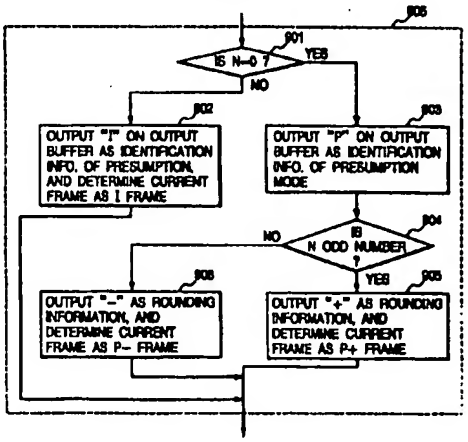
(54) Image sequence coding and decoding method

(57) In order to prevent accumulation of rounding errors caused by the integer-based bilinear interpolation used in the motion compensation process of image coding and decoding methods, two types of P frames are used for unidirectional motion compensation prediction, namely:

P+ frames, which round the results of bilinear interpolation with real number operations to the nearest integer and rounds half integer values (0.5 added to an integer) away from zero, and
P- frames, which differs from p+ frames in that the above mentioned half integer values are rounded towards zero.

Utilising both of these P+ and P- frames enables cancelling of rounding errors and prevents the accumulation of rounding errors.

FIG. 9



EP 0 884 912 A2

Description

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates to an image sequence coding and decoding method which performs interframe prediction using quantized values for chrominance or luminance intensity.

Description of Related Art

In high efficiency coding of image sequences, interframe prediction (motion compensation) by utilizing the similarity of adjacent frames over time, is known to be ahighly effective technique for data compression. Today's most frequently used motion compensation method is block matching with half pixel accuracy, which is used in international standards H.263, MPEG1, and MPEG2. In this method, the image to be coded is segmented into blocks and the horizontal and vertical components of the motion vectors of these blocks are estimated as integral multiples of half the distance between adjacent pixels. This process is described using the following equation:

[Equation 1]

$$P(x,y)=R(x+u_i,y+v_i)\ (x,y)\in B_i, 0\le i<N \tag{1}$$

where P(x, y) and R(x, y) denote the sample values (luminance or chrominance intensity) of pixels located at coordinates (x, y) in the predicted image P of the current frame and the reference image (decoded image of a frame which has been encoded before the current frame) R, respectively. x and y are integers, and it is assumed that all the pixels are located at points where the coordinate values are integers. Additionally it is assumed that the sample values of the pixels are quantized to non-negative integers. N, Bi, and (ui, vi) denote the number of blocks in the image, the set of pixels included in the i-th block of the image, and the motion vectors of the i-th block, respectively.

When the values for ui and vi are not integers, it is necessary to find the intensity value at the point where no pixels actually exist in the reference image. Currently, bilinear interpolation using the adjacent four pixels is the most frequently used method for this process. This interpolation method is described using the following equation:

[Equation 2]

$$R\left(x+\frac{p}{d},y+\frac{q}{d}\right)=((d-q)((d-p)R(x,y)+pR(x+1,y))+q((d-p)R(x,y+1)+pR(x+1,y+1)))//d^2 \tag{2}$$

where d is a positive integer, and p and q are smaller than d but not smaller than 0. "//" denotes integer division which rounds the result of normal division (division using real numbers) to the nearest integer.

An example of the structure of an H.263 video encoder is shown in Fig. 1. As the coding algorithm, H.263 adopts a hybrid coding method (adaptive interframe/intraframe coding method) which is a combination of block matching and DCT (discrete cosine transform). A subtractor 102 calculates the difference between the input image (current frame base image) 101 and the output image 113 (related later) of the interframe/intraframe coding selector 119 , and then outputs an error image 103. This error image is quantized in a quantizer 105 after being converted into DCT coefficients in a DCT converter 104 and then forms quantized DCT coefficients 106. These quantized DCT coefficients are transmitted through the communication channel while at the same time used to synthesize the interframe predicted image in the encoder. The procedure for synthesizing the predicted image is explained next. The above mentioned quantized DCT coefficients 106 forms the reconstructed error image 110 (same as the reconstructed error image on the receive side) after passing through a dequantizer 108 and inverse DCT converter 109. This reconstructed error image and the output image 113 of the interframe /intraframe coding selector 119 is added at the adder 111 and the decoded image 112 of the current frame (same image as the decoded image of current frame reconstructed on the receiver side) is obtained. This image is stored in a frame memory 114 and delayed for a time equal to the frame interval. Accordingly, at the current point, the frame memory 114 outputs the decoded image 115 of the previous frame. This decoded image of the previous frame and the original image 101 of the current frame are input to the block matching section 116 and block matching isperformed between these images In the block matching process, the original image of the current frame is segmented into multiple blocks, and the predicted image 117 of the current frame is synthesized by extracting the section most resembling these blocks from the decoded image of the previous frame. In this process, it is necessary to estimate the motion between the prior frame and the current frame for each block. The motion vector for each block estimated in the motion estimation process is transmitted to the receiver side as motion vector data 120. On the receiver side, the same prediction image as on the transmitter side is synthesized using the motion vector information and the

decoding image of the previous frame. The prediction image 117 is input along with a "0" signal 118 to the interframe /intraframe coding selector 119. This switch 119 selects interframe coding or intraframe coding by selecting either of these inputs. Interframe coding is performed when the prediction image 117 is selected (this case is shown in Fig. 2). On the other hand when the "0" signal is selected, intraframe coding is performed since the input image itself is converted, to a DCT coefficients and output to the communication channel. In order for the receiver side to correctly reconstruct the coded image, the reciever must be informed whether intraframe coding or interframe coding was performed on the transmitter side. Consequently, an identifier flag 121 is output to the communication circuit. Finally, an H.263 coded bitstream 123 is acquired by multiplexing the quantized DCT coefficients, motion vectors, the and interframe/intraframe identifier flag information in a multiplexer 122.

The structure of a decoder 200 for receiving the coded bit stream output from the encoder of Fig. 1 is shown in Fig. 2. The H.263 coded bit stream 217 that is received is demultiplexed into quantized DCT coefficients 201, motion vector data 202, and a interframe/intraframe identifier flag 203 in the demultiplexer 216. The quantized DCT coefficients 201 become a decoded error image 206 after being processed by an inverse quantizer 204 and inverse DCT converter 205. This decoded error image is added to the output image 215 of the Interframe /intraframe coding selector 214 in an adder 207 and the sum of these images is output as the decoded image 208. The output of the interframe /intraframe coding selector is switched according to the interframe/intraframe identifier flag 203. A prediction image 212 utilized when performing interframe encoding is synthesized in the prediction image synthesizer 211. In this synthesizer, the position of the blocks in the decoded image 210 of the prior frame stored in frame memory 209 is shifted according to the motion vector data 202. On the other hand, for intraframe coding, the interframe /intraframe coding selector outputs the "0" signal 213 as is.

SUMMARY OF THE INVENTION

The image encoded by H.263 is comprised of a luminance plane (Y plane) containing luminance information, and two chrominance planes (U plane and V plane) containing chrominance information. At this time, characteristically, when the image has 2m pixels in the horizontal direction and 2n pixels in the vertical direction (m and n are positive integers), the Y plane has 2m pixels horizontally and 2n pixels vertically, the U and V planes have m pixels horizontally and n pixels vertically. The low resolution on the chrominance plane is due to the fact that the human visual system has a comparatively dull visual faculty with respect to spatial variations in chrominance. Having such image as an input, H. 263 performs coding and decoding in block units referred to as macroblocks. The structure of a macroblock is shown in Fig. 3. The macroblock is comprised of three blocks; a Y block, U block and V block. The size of the Y block 301 containing the luminance information is 16 X 16 pixels, and the size of the U block 302 and V block 303 containing the chrominance information is 8 X 8 pixels.

In H. 263, half pixel accuracy block matching is applied to each block. Accordingly, when the estimated motion vector is defined as (u, v), u and v are both integral multiples of half the distance between pixels. In other words, 1/2 is used as the minimum unit. The configuration of the interpolation method used for the intensity values (hereafter the intensity values for "luminance" and "chrominance" are called by the general term "intensity value") is shown in Fig. 4. When performing the interpolation described in equation 2, the quotients of division are rounded off to the nearest integer, and further, when the quotient has a half integer value (i.e. 0.5 added to an integer), rounding off is performed to the next integer in the direction away from zero. In other words, in Fig. 4, when the intensity values for 401, 402, 403, 404 are respectively La, Lb, Lc, and Ld (La, Lb, Lc, and Ld are non-negative integers), the interpolated intensity values Ia, Ib, Ic, and Id (Ia, Ib. Ic, and Id are non-negative integers) at positions405, 406, 407, 408 are expressed by the following equation:

[Equation 3]

$$Ia = La \tag{3}$$

$$Ib = [(La+Lb+1)/2]$$

$$Ic = [(La+Lc+1)/2]$$

$$Id = [(La+Lb+Lc+Ld+2)/4]$$

where "[ ]" denotes truncation to the nearest integer towards 0 (i.e. the fractional part is discarded). The expectation of the errors caused by this rounding to integers is estimated as follows: It is assumed that the probability that the intensity value at positions 405, 406, 407, and 408 of Fig. 4 is used is all 25 percent. When finding the intensity value Ia for position 405, the rounding error will clearly be zero. Also, when finding the intensity value Ib for position 406, the error will be zero when La+Lb is an even number, and when an odd number the error is 1/2. If the probability that La+Lb will be

an even number and an odd number is both 50 percent, then the expectation for the error will be 0 x 1 / 2 + 1 / 2 X 1 / 2 = 1 / 4. Further, when finding the intensity value Ic for position 407, the expectation for the error is 1 / 4 as for Ib. When finding the intensity value Id for position 408, the error when the residual of La+Lb+Lc+Ld divided by four are 0, 1, 2, and 3 are respectively 0, -1/4, 1/2, and 1/4. If we assume that the probability that the residual is 0, 1, 2, and 3 is all equal (i.e. 25 percent), the expectation for the error is 0 x 1 / 4 - 1 / 4 x 1 / 4 + 1 / 2 x 1 / 4 + 1 / 4 x 1 / 4 = 1 / 8. As described above, assuming that the possibility that the intensity value at positions 405 - 408 being used are all equal, the final expectation for the error is 0 x 1 / 4 + 1 / 4 x 1 / 4 + 1 / 4 x 1 / 4 + 1 / 8 x 1 / 4 = 5 / 32. This indicates that each time motion compensation is performed by means of block matching, an error of 5/32 occurs in the pixel intensity value. Generally in low rate coding, sufficient number of bits cannot be used for the encoding of the interframe error difference so that the quantized step size of the DCT coefficient is prone to be large. Accordingly, errors occurring due to motion compensation are corrected only when it is very large. When interframe encoding is performed continuously without performing intraframe coding under such environment, the errors tend. to accumulate and cause bad effects on the reconstructed image.

Just as explained above, the number of pixels is about half in both the vertical and horizontal direction on the chrominance plane Therefore, for the motion vectors of the U block and V block, half the value of the motion vector for the Y block is used for the vertical and horizontal components. Since the horizontal and vertical components of the motion vector for the Y block motion vector are integral multiples of 1/2, the motion vector components for the U and V blocks will appear as integral multiples of 1/4 (quarter pixel accuracy) if ordinary division is implemented. However, due to the high computational complexity of the intensity interpolation process for motion vectors with quarter pixel accuracy, the motion vectors for U and V blocks are rounded to half pixel accuracy in H.263. The rounding method utilized in H.263 is as follows: According to the definition described above, (u, v) denotes the motion vector of the macroblock (which is equal to the motion vector for the Y block). Assuming that r is an integer and s is an non-negative integer smaller than 4, u / 2 can be rewritten as u / 2 = r + s / 4 . When s is 0 or 2, no rounding is required since u / 2 is already an integral multiple of 1 / 2. However when s is equal to 1 or 3, the value of s is rounded to 2. By increasing the possibility that s takes the value of 2 using this rounding method, the filtering effect of motion compensation can be emphasized. When the probability that the value of s prior to rounding is 0, 1, 2, and 3 are all 25 percent, the probability that s will be 0 or 2 after rounding will respectively be 25 percent and 75 percent. The above explained process related to the horizontal component u of the motion vector is also applied to the vertical component v. Accordingly, in the U block and V block, the probability for using the intensity value of the 401 position is 1 / 4 X 1 / 4 = 1 / 16, and the probability for using the intensity value of the 402 and 403 positions is both 1 / 4 X 3 / 4 = 3 / 16, while the probability for using the intensity value of position 404 is 3 / 4 X 3 / 4 = 9 / 16. By utilizing the same method as above, the expectation for the error of the intensity value is 0 X 1 / 16 + 1 / 4 X 3 / 16 + 1 / 4 X 3 / 16 + 1 / 8 X 9 / 16 = 21 / 128. Just as explained above for the Y block, when interframe encoding is continuously performed, the problem of accumulated errors occurs.

As related above, for image sequence coding and decoding methods in which interframe prediction is performed and luminance or chrominance intensity is quantized, the problem of accumulated rounding errors occurs. This rounding error is generated when the luminance or chrominance intensity value is quantized during the generation of the interframe prediction image.

In view of the above problems, it is therefore an object of this invention, to improve the quality of the reconstructed image by preventing error accumulation.

In order to achieve the above object, the accumulation of errors is prevented by limiting the occurrence of errors or performing an operation to cancel out errors that have occurred.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram showing the layout of the H.263 image encoder.

Figure 2 is a block diagram showing the layout of the H.263 image decoder.

Figure 3 is a drawing showing the structure of the macro block.

Figure 4 is a drawing showing the interpolation process of intensity values for block matching with half pixel accuracy.

Figure 5 is a drawing showing a coded image sequence.

Figure 6 is a block diagram showing a software image encoding device.

Figure 7 is a block diagram showing a software image decoding device.

Figure 8 is a flow chart showing an example of processing in the software image encoding device.

Figure 9 is a flow chart showing an example of the coding mode decision processing for the software image encoding device.

Figure 10 is a flow chart showing an example of motion estimation and motion compensation processing in the software image encoding device.

Figure 11 is a flow chart showing the processing in the software image decoding device.

Figure 12 is a flow chart showing an example of motion compensation processing in the software image decoding device.

Figure 13 is a drawing showing an example of a storage media on which an encoded bit stream generated by an encoding method that outputs bit streams including I, P+ and P- frames is recorded.

Figure 14 is a set of drawings showing specific examples of devices using an encoding method where P+ and P- frames coexist.

Figure 15 is a drawing showing an example of a storage media on which an encoded bit stream generated by an encoding method the outputs bit streams including I, B, P+, and P-frames is recorded.

Figure 16 is a block diagram showing an example of a block matching unit included in a device using an encoding method where P+ and P- frames coexist.

Figure 17 is a block diagram showing the prediction image synthesizer included in a device for decoding bit streams encoded by an encoding method where P+ and P- frames coexist.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

First, in which circumstances the accumulated rounding errors as related in the "Prior art" occur must be considered. An example of an image sequences encoded by coding methods which can perform both unidirectional prediction and bidirectional prediction such as in MPEG. 1, MPEG. 2 and H. 263 is shown in Fig. 5. An image 501 is a frame-coded by means of intraframe coding and is referred to as an I frame. In contrast, images 503, 505, 507, 509 are called P frames and are coded by unidirectional interframe coding by using the previous I or P frame as the reference image. Accordingly, when for instance encoding image 505, image 503 is used as the reference image and interframe prediction is performed. Images 502, 504, 506 and 508 are called B frames and bidirectional interframe prediction is performed utilizing the previous and subsequent I or P frame. The B frame is characterized by not being used as a reference image when interframe prediction is performed. Since motion compensation is not performed in I frames, the rounding error caused by motion compensation will not occur. In contrast, not only is motion compensation performed in the P frames but the P frame is also used as a reference image by other P or B frames so that it may be a cause leading to accumulated rounding errors. In the B frames on the other hand, motion compensation is performed so that the effect of accumulated rounding errors appears in the reconstructed image. However, due to the fact that B frames are not used as reference images, B frames cannot be a source of accumulated rounding errors. Thus, if accumulated rounding errors can be prevented in the P frame, then the bad effects of rounding errors can be alleviated in the overall image sequence. In H.263 a frame for coding a P frame and a B frame exists and is called a PB frame (For instance, frames 503 and 504 can both be encoded as a PB frame.) If the combined two frames are viewed as separate frames, then the same principle as above can be applied. In other words, if countermeasures are taken versus rounding errors for the P frame part within a PB frame, then the accumulation of errors can be prevented.

Rounding errors occur during interpolation of intensity values when a value obtained from normal division (division whose operation result is a real number) is a half integer (0.5 added to an integer) and this result is then rounded up to the next integer in the direction away from zero. For instance, when dividing by 4 to find an interpolated intensity value is performed, the rounding errors for the cases when the residual is 1 and 3 have equal absolute values but different signs. Consequently, the rounding errors caused by these two cases are canceled when the expectation for the rounding errors is calculated (in more general words, when dividing by a positive integer d' is performed, the rounding errors caused by the cases when the residual is t and d' -t are cancelled). However, when the residual is 2, In other words when the result of normal division is a half integer, the rounding error cannot be canceled and leads to accumulated errors. To solve this problem, a method that allows the usage of two rounding methods can be used. The two rounding methods used here are: a rounding method that rounds half integers away from 0; and a rounding method that rounds half integers towards 0. By combining the usage of these two rounding methods, the rounding errors can be canceled. Hereafter, the rounding method that rounds the result of normal division to the nearest integer and rounds half integer values away from 0 is called "positive rounding". Additionally, the rounding method that rounds the result of normal division to the nearest integer and rounds half integer values towards 0 is called "negative rounding". The process of positive rounding used in block matching with half pixel accuracy is shown in Equation 3. When negative rounding is used instead, this equation can be rewritten as shown below.

[Equation 4]

$$Ia = La \tag{4}$$

$$Ib = [(La+Lb)/2]$$

$$Ic = [(La+Lc)/2]$$

$$Id = [(La+Lb+Lc+Ld+1)/4]$$

Hereafter motion compensation methods that performs positive and negative rounding for the synthesis of inter-frame prediction images are called "motion compensation using positive rounding" and "motion compensation using negative rounding", respectively. Furthermore, for P frames which use block matching with half pixel accuracy for motion compensation, a frame that uses positive rounding is called a "P+ frame" and a frame that uses negative rounding is called a "P- frame" (under this definition, the P frames in H. 263 are all P+ frames). The expectation for the rounding errors in P+ and P- frames have equal absolute values but different signs. Accordingly, the accumulation of rounding errors can be prevented when P+ frames and P- frames are alternately located along the time axis. In the example in Fig. 5, if the frames 503 and 507 are set as P+ frames and the frames 505 and 509 are set as P- frames, then this method can be implemented. The alternate occurrence of P+ frames and P- frames leads to the usage of a P+ frame and a P- frame in the bidirectional prediction for B frames. Generally, the average of the forward prediction image (i.e. the prediction image synthesized by using frame 503 when frame 504 in Fig. 5 is being encoded) and the backward prediction image (i.e. the prediction image synthesized by using frame 505 when frame 504 in Fig. 5 is being encoded) is frequently used for synthesizing the prediction image for B frames. This means that using a P+ frame (which has a positive value for the expectation of the rounding error) and a P- frame (which has a negative value for the expectation of the rounding error) in bidirectional prediction for a B frame is effective in canceling out the effects of rounding errors. Just as related above, the rounding process in the B frame will not be a cause of error accumulation. Accordingly, no problem will occur even if the same rounding method is applied to all the B frames. For instance, no serious degradation of decoded images is caused even if motion compensation using positive rounding is performed for all of the B frames 502, 504, 506, and 508 in Fig. 5. Preferably only one type of rounding is performed for a B frame, in order to simplify the B frame decoding process.

A block matching section 1600 of an image encoder according-to the above described motion compensation method utilizing multiple rounding methods is shown in Fig. 16. Numbers identical to those in other drawings indicate the same part. By substituting the block matching section 116 of Fig. 1 with 1600, multiple rounding methods can be used. Motion estimation processing between the input image 101 and the decoded image of the previous frame is performed in a motion estimator 1601. As a result, motion information 120 is output. This motion information is utilized in the synthesis of the prediction image in a prediction image synthesizer 1603. A rounding method determination device 1602 determines whether to use positive rounding or negative rounding as the rounding method for the frame currently being encoded. Information 1604 relating to the rounding method that was determined is input to the prediction image synthesizer 1603. In this prediction image synthesizer 1603, a prediction image 117 is synthesized and output based on the rounding method determined by means of information 1604. In the block matching section 116 in Fig. 1, there are no items equivalent to 1602, 1604 of Fig. 16, and the prediction image is synthesized only by positive rounding. Also, the rounding method 1605 determined at the block matching section can be output, and this information can then be multiplexed into the bit stream and be transmitted.

A prediction image synthesizer 1700 of an image decoder which can decode bit streams generated by a coding method using multiple rounding methods is shown in Fig. 17. Numbers identical to those in other drawings indicate the same part. By substituting the prediction image synthesizer 211 of Fig. 2 by 1700, multiple rounding methods can be used. In the rounding method determination device 1701, the rounding method appropriate for prediction image synthesis in the decoding process is determined. In order to carry out decoding correctly, the rounding method selected here must be the same as the rounding method that was selected for encoding. For instance the following rule can be shared between the encoder and decoder: When the current frame is a P frame and the number of P frames (including the current frame) counted from the most recent I frame is odd, then the current frame is a P+ frame. When this number is even, then the current frame is a P- frame. If the rounding method determination device on the encoding side (For instance, 1602 in Fig. 16) and the rounding method determination device 1701 conform to this common rule, then the images can correctly be decoded. The prediction image is synthesized in the prediction image synthesizer 1703 using motion information 202, decoding image 210 of the prior frame, and information 1702 related to the rounding method determined as just described. This prediction image 212 is output and then used for the synthesis of the decoded image. As an alternative to the above mentioned case, a case where the information related to the rounding method is multiplexed in the transmitted bit stream can also be considered (such bit stream can be generated at the encoder by outputting the information 1605 related to the rounding method from the block matching section depicted in Fig. 16). In such case, the rounding method determiner device 1701 is not used, and information 1704 related to the rounding method extracted from the encoded bit stream is used at the prediction image synthesizer 1703.

Besides the image encoder and the image decoder utilizing the custom circuits and custom chips of the conventional art as shown in Fig. 1 and Fig. 2, this invention can also be applied to software image encoders and software image decoders utilizing general-purpose processors. A software image encoder 600 and a software image decoder 700 are shown in Fig. 6 and Fig. 7. In the software image encoder 600, an input image 601 is first stored in the input frame memory 602 and the general-purpose processor 603 loads information from here and performs encoding. The

program for driving this general-purpose processor is loaded from a storage device 608 which can be a hard disk, floppy disk, etc. and stored in a program memory 604. This general-purpose processor also uses a process memory 605 to perform the encoding. The encoding information output by the general-purpose processor is temporarily stored in the output buffer 606 and then output as an encoded bit stream 607.

A flowchart for the encoding software (recording medium readable by computer) is shown in Fig. 8. The process starts in 801, and the value 0 is assigned to variable N in 802. Next, in 803 and 804, the value 0 is assigned to N when the value for N is 100. N is a counter for the number of frames. 1 is added for each one frame whose processing is complete, and values from 0 to 99 are allowed when performing coding. When the value for N is 0, the current frame is an I frame. When N is an odd number, the current frame is a P+ frame, and when an even number other than 0, the current frame is a P- frame. When the upper limit for the value of N is 99, it means that one I frame is coded after 99 P frames (P+ frames or P- frames) are coded. By always inserting one I frame in a certain number of coded frames, the following benefits can be obtained: (a) Error accumulation due to a mismatch between encoder and decoder processing can be prevented (for instance, a mismatch in the computation of DCT); and (b) The processing load for acquiring the reproduced image of the target frame from the coded data (random access) is reduced. The optimal N value varies when the encoder performance or the environment where the encoder is used are changed. It does not mean, therefore, that the value of N must always be 100. The process for determining the rounding method and coding mode for each frame is performed in 805 and the flowchart with details of this operation is shown in Fig. 9. First of all, whether N is a 0 or not is checked in 901. If N is 0, then 'I' is output as distinction information of the prediction mode, to the output buffer in 902. This means that the image to be coded is will be coded as an I frame. Here, "output to the output buffer" means that after being stored in the output buffer, the information is output to an external device as a portion of the coded bit stream. When N is not 0, then whether N is an odd or even number is identified in 904. When N is an odd number, '+' is output to the output buffer as the distinction information for the rounding method in 905, and the image to be coded will be coded as a P+ frame. On the other hand, when N is an even number, '-' is output to the output buffer as the distinction information for the rounding method in 906, and the image to be coded will be coded as a P- frame. The process again returns to Fig. 8, where after determining the coding mode in 805, the input image is stored in the frame memory A in 806. The frame memory A referred to here signifies a portion of the memory zone (for instance, the memory zone maintained in the memory of 605 in Fig. 6) of the software encoder. In 807, it is checked whether the frame currently being coded is an I frame. When not identified as an I frame, motion estimation and motion compensation is performed in 808. The flowchart in Fig. 10 shows details of this process performed in 808. First of all, in 1001, motion estimation is performed between the images stored in frame memories A and B (just as written in the final part of this paragraph, the decoded image of the prior frame is stored in frame memory B). The motion vector for each block is found, and this motion vector is sent to the output buffer. Next, in 1002, whether or not the current frame is a P+ frame is checked. When the current frame is a P+ frame, the prediction image is synthesized in 1003 utilizing positive rounding and this prediction image is stored in frame memory C. On the other hand, when the current frame is a P- frame, the prediction image is synthesized in 1004 utilizing negative rounding and this prediction image is stored in the frame memory C. Next, in 1005, the differential image between frame memories A and C is found and stored in frame memory A. Here, the process again returns to Fig. 8. Prior to starting the processing in 809, the input image is stored in frame memory A when the current frame is an I frame, and the differential image between the input image and the prediction image is stored in frame memory A when the current frame is a P frame (P+ or P- frame).. In 809, DCT is applied to the image stored in frame memory A, and the DCT coefficients calculated here are sent to the output buffer after being quantized. In 810, inverse quantization is performed to the quantized DCT coefficients and inverse DCT is applied. The image obtained by applying inverse DCT is stored in frame memory B. Next in 811, it is checked again whether the current frame is an I frame. When the current frame is not an I frame, the images stored in frame memory B and C are added and the result is stored in frame memory B. The coding process of a frame ends here, and the image stored in frame memory B before going into 813 is the reconstructed image of this frame (this image is identical with the one obtained at the decoding side). In 813, it is checked whether the frame whose coding has just finished is the final frame in the sequence. If this is true, the coding process ends . If this frame is not the final frame, 1 is added to N in 814, and the process again returns to 803 and the coding process for the next frame starts.

A software decoder 700 is shown in Fig. 7. After the coded bit stream 701 is temporarily stored in the input buffer 702, this bit stream is then loaded into the general-purpose processor 703. The program for driving this general-purpose processor is loaded from a storage device 708 which can be a hard disk, floppy disk, etc. and stored in a program memory 704. This general-purpose processor also uses a process memory 605 to perform the decoding. The decoded image obtained by the decoding process is temporarily stored in the output frame memory 706 and then sent out as the output image 707.

A flowchart of the decoding software for the software decoder 700 shown in Fig. 7 is shown in Fig. 11. The process starts in 1101, and it is checked in 1102 whether input information is present. If there is no input information, the decoding process ends in 1103. When input information is present, distinction information of the prediction mode is input in 1104. The word "input" used here means that the information stored in the input buffer (for instance 702 of Fig. 7) is

loaded by the general-purpose processor. In 1105, it is checked whether the encoding mode distinction information is "I". When not "I", the distinction information for the rounding method is input and synthesis of the interframe prediction image is performed in 1107. A flowchart showing details of the operation in 1107 is shown in Fig. 12. In 1201, a motion vector is input for each block. Then, in 1202, it is checked whether the distinction information for the rounding method loaded in 1106 is a "+". When this information is "+", the frame currently being decoded is a P+ frame. In this case, the prediction image is synthesized using positive rounding in 1203, and the prediction image is stored in frame memory D. Here, frame memory D signifies a portion of the memory zone of the software decoder (for instance, this memory zone is obtained in the processing memory 705 in Fig. 7). When the distinction information of the rounding method is not "+", the current frame being decoded is a P- frame. The prediction image is synthesized using negative rounding in 1204 and this prediction image is stored in frame memory D. At this point, if a P+ frame is decoded as a P- frame due to some type of error, or conversely if a P- frame is decoded as a P+ frame, the correct prediction image is not synthesized in the decoder and the quality of the decoded image deteriorates. After synthesizing the prediction image, the operation returns to Fig. 11 and the quantized DCT coefficients is input in 1108. Inverse quantization and inverse DCT is then applied to these coefficients and the resulting image is stored in frame memory E. In 1109, it is checked again whether the frame currently being decoded is an I frame. If the current frame is not an I frame, images stored in frame memory D and E are added in 1110 and the resulting sum image is stored in frame memory E. The image stored in frame memory E before starting the process in 1111 is the reconstructed image. This image stored in frame memory E is output to the output frame memory (for instance, 706 in Fig. 7) in 1111, and then output from the decoder as the reconstructed image. The decoding process for a frame is completed here and the process for the next frame starts by returning to 1102.

When a software based on the flowchart shown in Figs. 8 - 12 is run in the software image encoders or decoders, the same effect as when custom circuits and custom chips are utilized are obtained.

A storage media (recording media) with the bit stream generated by the software encoder 601 of Fig. 6 being recorded is shown in Fig. 13. It is assumed that the algorithms shown in the flowcharts of Figs. 8 - 10 is used in the software encoder. Digital information is recorded concentrically on a recording disk 1301 capable of recording digital information (for instance magnetic disks, optical disk, etc.). A portion 1302 of the information recorded on this digital disk includes: prediction mode distinction information 1303, 1305, 1308, 1311, and 1314; rounding method distinction information 1306, 1309, 1312, and 1315; and motion vector and DCT coefficient information 1304, 1307, 1310, 1313, and 1316. Information representing 'I' is recorded in 1303, 'P' is recorded in 1305, 1308, 1311, and 1314, '+' is recorded in 1306, and 1312, and '-' is recorded in 1309, and 1315. In this case, 'I' and '+' can be represented by a single bit of 0, and 'P' and '-' can be represented by a single bit of 1. Using this representation, the decoder can correctly interpret the recorded information and the correct reconstructed image is synthesized. By storing a coded bit stream in a storage media using the method described above, the accumulation of rounding errors is prevented when the bit stream is read and decoded.

A storage media with the bit stream of the coded data of the image sequence shown in Fig. 5 being recorded is shown in Fig. 15. The recorded bit stream includes information related to P+, P-, and B frames. In the same way as in 1301 of Fig. 13, digital information is recorded concentrically on a record disk 1501 capable for recording digital information(for instance, magnetic disks, optical disks, etc.). A portion 1502 of the digital information recorded on this digital disk includes: prediction mode distinction information 1503, 1505, 1508, 1510, and 1513; rounding method distinction information 1506, and 1512; and motion vector and DCT coefficient information 1504, 1507, 1509, 1511, and 1514. Information representing 'I' is recorded in 1503, 'P' is recorded in 1505, and 1510, 'B' is recorded in 1508, and 1513, '+' is recorded in 1505, and '-' is recorded in 1511. In this case, 'I', 'P' and 'B' can be represented respectively by two bit values 00, 01, and 10, and '+' and '-' can be represented respectively by one bit values 0 and 1. Using this representation, the decoder can correctly interpret the recorded information and the correct reconstructed is synthesized. In Fig. 15, information related to frame 501 (I frame) in Fig. 5 is 1503 and 1504, information related to 502 (B frame) is 1508 and 1509, information related to frame 503 (P+ frame) is 1505 and 1507, information related to frame 504 (B frame) is 1513 and 1514, and information related to frame 505 (P- frame) is 1510 and 1512. When coding image sequences are coded using B frames, the transmission order and display order of frames are usually different. This is because the previous and subsequent reference images need to be coded before the prediction image for the B frame is synthesized. Consequently, in spite of the fact that the frame 502 is displayed before frame 503, information related to frame 503 is transmitted before information related to frame 502. As described above, there is no need to use multiple rounding methods for B frames since motion compensation in B frames do not cause accumulation of rounding errors. Therefore, as shown in this example, information that specifies rounding methods (e.g. '+' and '-') is not transmitted for B frames. Thus for instance, even if only positive rounding is applied to B frames, the problem of accumulated rounding errors does not occur. By storing coded bit streams containing information related to B frames in a storage media in the way described above, the occurrence of accumulated rounding errors can be prevented when this bit stream is read and decoded.

Specific examples of coders and decoders using the coding method described in this specification is shown in Fig.

14. The image coding and decoding method can be utilized by installing image coding and decoding software into a computer 1401. This software is recorded in some kind of storage media (CD-ROM, floppy disk, hard disk, etc.) 1412, loaded into a computer and then used. Additionally, the computer can be used as an image communication terminal by connecting the computer to a communication lines. It is also possible to install the decoding method described in this specification into a player device 1403 that reads and decodes the coded bit stream recorded in a storage media 1402. In this case, the reconstructed image signal can be displayed on a television monitor 1404. The device 1403 can be used only for reading the coded bit stream, and in this case, the decoding device can be installed in the television monitor 1404. It is well known that digital data transmission can be realized using satellites and terrestrial waves. A decoding device can also be installed in a television receiver 1405 capable of receiving such digital transmissions. Also, a decoding device can also be installed inside a set top box 1409 connected to a satellite/terrestrial wave antenna, or a cable 1408 of a cable television system, so that the reconstructed images can be displayed on a television monitor 1410. In this case, the decoding device can be incorporated in the television monitor rather than in the set top box, as in the case of 1404. The layout of a digital satellite broadcast system is shown in 1413, 1414 and 1415. The video information in the coded bit stream is transmitted from a broadcast station 1413 to a communication or broadcast satellite 1414. The satellite receives this information, sends it to a home 1415 having equipment for receiving satellite broadcast programs, and the video information is reconstructed and displayed in this home using devices such as a television receiver or a set top box. Digital image communication using mobile terminals 1406 has recently attracted considerable attention, due to the fact that image communication at very low bit rates has become possible. Digital portable terminals can be categorized in the following three types: a transceiver having both an encoder and decoder; a transmitter having only an encoder; and a receiver having only a decoder. An encoding device can be installed in a video camera recorder 1407. The camera can also be used just for capturing the video signal and this signal can be supplied to a custom encoder 1411. All of the devices or systems shown in this drawing can be equipped with the coding or/and decoding method described in this specification. By using this coding or/and decoding method in these devices or systems, images of higher quality compared with those obtained using conventional technologies can be obtained.

The following variations are clearly included within the scope of this invention.

(i) A prerequisite of the above described principle was the use of block matching as a motion compensation method. However, this invention is further capable of being applied to all image sequence coding and decoding methods in which motion compensation is performed by taking a value for the vertical and horizontal components of the pixel motion vector that is other than an integer multiple of the sampling period in the vertical and horizontal directions of the pixel, and then finding by interpolation, the intensity value of a position where the sample value is not present. Thus for instance, the global motion compensation listed in Japanese Patent Application No. Hei 08-060572 and the warping prediction listed in Japanese Patent Application No. Hei 08-249601 are applicable to the method of this invention.

(ii) The description of the invention only mentioned the case where a value integral multiple of 1/2 was taken for the horizontal and vertical components of the motion vector. However, this invention is also generally applicable to methods in which integral multiples of 1/d (d is a positive integer and also an even number) are allowed for the horizontal and vertical components of the motion vector. However, when d becomes large, the divisor for division in bilinear interpolation (square of d, see Equation 2) also becomes large, so that in contrast, the probability of results from normal division reaching a value of 0.5 become low. Accordingly, when performing only positive rounding, the absolute value of the expectation for rounding errors becomes small and the bad effects caused by accumulated errors become less conspicuous. Also applicable to the method of this invention, is a motion compensation method where for instance, the d value is variable, both positive rounding and negative rounding are used when d is smaller than a fixed value, and only positive rounding or only negative rounding is used when the value of d is larger than a fixed value.

(iii) As mentioned in the prior art, when DCT is utilized as an error coding method, the adverse effects from accumulated rounding errors are prone to appear when the quantized step size of the DCT coefficient is large. However a method is also applicable to the invention, in which, when the quantization step size of DCT coefficients is larger than a threshold value then both positive rounding and negative rounding are used. When the quantization step size of the DCT coefficients is smaller than the threshold value then only positive rounding or only negative rounding is used.

(iv) In cases where error accumulations occur on the luminance plane and cases where error accumulations occur on the chrominance plane, the bad effects on the reconstructed images are generally more serious in the case of error accumulations on the chrominance plane. This is due to the fact that rather than cases where the image darkens or lightens slightly, cases where overall changes in the image color happen are more conspicuous. However, a method is also applicable to this invention in which both positive rounding and negative rounding are used for the chrominance signal, and only positive rounding or negative rounding is used for the luminance signal.

As described in the description of related art, 1/4 pixel accuracy motion vectors obtained by halving the 1/2

pixel accuracy motion vectors are rounded to 1/2 pixel accuracy in H.263. However by adding certain changes to this method, the absolute expectation value for rounding errors can be reduced. In H. 263 that was mentioned in the prior art, a value which is half the horizontal or vertical components of the motion vector for the luminance plane is expressed as r + s / 4 (r is an integer, s is an integer less than 4 and not smaller than 0), and when s is 1 or 3, a rounding operation is performed to obtain a 2. This operation can be changed as follows: When s is 1, a rounding operation is performed to obtain a 0, and when s is 3 a 1 is be added to r to make s a 0. By performing these operations, the number of times that the intensity values at positions 406 - 408 in Fig. 4 is definitely reduced (Probability that horizontal and vertical components of motion vector will be an integer become high.) so that the absolute expectation value for the rounding error becomes small. However, even if the size of the error occurring in this method can be limited, the accumulation of errors cannot be completely prevented.

(v) The invention described in this specification is applicable to a method that obtains the final interframe prediction image by averaging the prediction images obtained by different motion compensation methods. For example, in the method described in Japanese Patent Application No. Hei 8-2616, interframe prediction images obtained by the following two methods are averaged: block matching in which a motion vector is assigned to each 16x16 pixel block; and block matching in which a motion vector is assigned to each 8x8 pixel blocks. In this method, rounding is also performed when calculating the average of the two prediction images. When only positive rounding is continuously performed in this averaging operation, a new type of rounding error accumulates. This problem can be solved by using multiple rounding methods for this averaging operation. In this method, negative rounding is performed in the averaging operation when positive rounding is performed in block matching. Conversely, positive rounding is used for the averaging when negative rounding is used for block matching. By using different rounding methods for averaging and block matching, the rounding errors from two different sources is cancelled within the same framed..

(vi) When utilizing a method that alternately locates P+ frames and P- frames along the time axis, the encoder or the decoder needs to determine whether the currently processed P frame is a P+ frame or a P- frame. The following is an example of such identification method: A counter counts the number of P frames after the most recently coded or decoded I frame, and the current P frame is a P+ frame when the number is odd, and a P- frame when the number is even (this method is referred to as an implicit scheme). There is also a method for instance, that writes into the header section of the coded image information, information to identify whether the currently coded P frame at the encoder is a P+ frame or a P- frame (this method is referred to as an explicit scheme). Compared with the implicit method, this method is well able to withstand transmission errors, since there is no need to count the number of P frames.

Additionally, the explicit method has the following advantages: As described in "Description for Related Art", past encoding standards (such as MPEG-1 or MPEG-2) use only positive rounding for motion compensation. This means for instance that the motion estimation/motion compensation devices (for example equivalent to 106 in Fig.1) for MPEG-1/MPEG-2 on the market are not compatible with coding methods that uses both P+ frames and P- frames. It is assumed that there is a decoder which can decode bit streams generated by a coding method that uses P+ frames and P- frames. In this case if the decoder is based on the above mentioned implicit method, then it will be difficult to develop an encoder that generates bit streams that can be correctly decoded by the above mentioned decoder, using the above mentioned motion estimation/compensation device for MPEG-1/MPEG-2. However, if the decoder is based on the above mentioned explicit method, this problem can be solved. An encoder using an MPEG-1/MPEG-2 motion estimation/motion compensation device can continuously send P+ frames, by continuously writing rounding method distinction information indicating positive rounding into the frame information header. When this is performed, a decoder based on the explicit method can correctly decode the bit stream generated by this encoder. Of course, it should be more likely in such case that the accumulation of rounding errors occurs, since only P+ frames are present. However, error accumulation is not a serious problem in cases where the encoder uses only small values as the quantization step size for the DCT coefficients (an example for such coders is a custom encoder used only for high rate coding). In addition to this interoperability between past standards, the explicit method further have the following advantages: (a) the equipment cost for high rate custom encoders and coders not prone to rounding error accumulation due to frequent insertion of I frames can be reduced by installing only positive or negative rounding as the pixel value rounding method for motion compensation; and(b) the above encoders not prone to rounding error accumulation have the advantage in that there is no need to decide whether to code the current frame as a P+ or P- frame, and the processing is simplified.

(vii) The invention described in this specification is applicable to coding and decoding methods that applies filtering accompanying rounding to the interframe prediction images. For instance, in the international standard H. 261 for image sequence coding, a low-pass filter (called a loop filter) is applied to block signals whose motion vectors are not 0 in interframe prediction images. Also, in H. 263, filters can be used to smooth out discontinuities on block boundaries (blocking artifacts). All of these filters perform weighted averaging to pixel intensity values and rounding is then performed on the averaged intensity values. Even for these cases, selective use of positive rounding and negative rounding is effective for preventing error accumulation.

(viii) Besides I P+ P- P+ P- ..., various methods for mixing P+ frames and P- frames such as I P+ P+ P- P- P+ P+ ..., or I P+ P- P- P+ P+ ... are applicable to the method of this, invention. For instance, using a random number generator that outputs 0 and 1 both at a probability of 50 percent, the encoder can code a P+ and P- frame when the output is 0 and 1, respectively. In any case, the less the difference in probability that P+ frames and P- frames occur in a certain period of time, the less the rounding error accumulation is prone to occur. Further, when the encoder is allowed to mix P+ frames and P- frames by an arbitrary method, the encoder and decoder must operate based on the explicit method and not with the implicit method described above. Accordingly, the explicit method is superior when viewed from the perspective of allowing flexibility configuration for the encoder and decoder.

(ix) The invention described in this specification does not limit the pixel value interpolation method to bilinear interpolation. Interpolation methods for intensity values can generally be described by the following equation:

[Equation 5]

$$R(x+r,y+s)=T(\sum_{j=-x}^{x}\sum_{k=-x}^{x}h(r-j,s-k)R(x+j,y+k)) \tag{5}$$

where, r and s are real numbers, h(r, s) is a function for interpolating the real numbers, and T(z) is a function for rounding the real number z. The definitions of R (x, y), x, and y are the same as in Equation 4. Motion compensation utilizing positive rounding is performed when T (z) is a function representing positive rounding, and motion compensation utilizing negative rounding is performed when the function representing negative rounding. This invention is applicable to interpolation methods that can be described using Equation 5. For instance, bilinear interpolation can be described by defining h(r, s) as shown below.

[Equation 6]

$$h(r,s)=\begin{cases}(1-|r|)(1-|s|), & 0\le|r|\le 1, 0\le|s|\le 1,\\ 0, & otherwise.\end{cases} \quad \cdots(6)$$

However, if for instance h(r,s) is defined as shown below,

[Equation 7]

$$h(r,s)=\begin{cases}1-|r|-|s|, & 0\le|r|+|s|\le 1, rs<0,\\ 1-|r|, & |r|\ge|s|,\ |r|\le 1, rs\ge 0,\\ 1-|s|, & |s|>|r|,\ |s|\le 1, rs>0,\\ 0, & otherwise.\end{cases} \quad \cdots(7)$$

then an interpolation method different from bilinear interpolation is implemented but the invention is still applicable.

(x) The invention described in this specification does not limit the coding method for error images to DCT (discrete cosine transform). For instance, wavelet transform (for example, M. Antonioni, et. al, "Image Coding Using Wavelet Transform" IEEE Trans. Image Processing, vol. 1, no.2, April 1992) and Walsh-Hadamard transform (for example, A. N. Netravalli and B. G. Haskell, "Digital Pictures", Plenum Press, 1998) are also applicable to this invention.

Claims

1. An image sequence coding method comprising the steps of:

synthesising a prediction image by utilising motion compensation, and
multiplexing information related to the difference image between an input image and said prediction image, and information related to motion vectors estimated during said motion compensation,
wherein there is a case where said prediction image is synthesised by motion compensation using pos-

itive rounding, and a case where said prediction image is synthesised by motion compensation using negative rounding.

2. The method of claim 1, wherein information related to the rounding method used for the synthesis of said prediction image is multiplexed with said information related to said difference image and said information related to motion vectors.

3. An image sequence coding method comprising the steps of:

synthesising a prediction image by utilising motion compensation, and
multiplexing information related to the difference image between an input image and said prediction image, and information related to motion vectors estimated during said motion compensation,
wherein information related to the rounding method used for the synthesis of said prediction image is multiplexed with said information related to said difference image and said information related to motion vectors.

4. An image sequence coding method comprising the steps of:

synthesising a prediction image by utilising motion compensation between an input image and a reference image, and
multiplexing information related to the difference image between said input image and said prediction image, and information related to motion vectors estimated during said motion compensation, wherein:
said motion compensation has half pixel accuracy and the intensity values of chrominance or luminance at a point on the reference image where no pixels are present are calculated by bilinear interpolation, and
there is a case where said bilinear interpolation is calculated by positive rounding according to $Ib = [(La+Lb+1)/2]$, $Ic = [(La+Lc+1)/2]$, and $Id = [(La+Lb+Lc+Ld+2)/4]$, and a case where said bilinear interpolation is calculated by negative rounding according to $Ib = [(La+Lb)/2]$, $Ic = [(La+Lc)/2]$, and $Id = [(La+Lb+Lc+Ld+1)/4]$, where La, Lb, Lc, and Ld are respectively the intensity values of a first pixel, a second pixel which is horizontally adjacent to said first pixel, a third pixel which is vertically adjacent to said first pixel, and a forth pixel which is vertically adjacent to said second pixel and horizontally adjacent to said third pixel, and Lb, Ic, and Id are respectively the interpolated intensity values of the middle point between said first and second pixel, the middle point between said first and third pixel, and the middle point between said first, second, third, and fourth pixel.

5. An image sequence coding method for encoding each frame of an image sequence consisting of plurality of frames, comprising the steps of:

synthesising the prediction image of a current frame from the decoded image of a previously encoded frame and the input image of said current frame by means of a first motion compensation,
generating information related to the difference image of said prediction image and the input image of said current frame,
multiplexing and then outputting information related to said difference image and information related to motion vectors estimated by said first motion compensation,
synthesising the decoded image of said current frame by utilising information related to said difference image and prediction image of said current frame, and
synthesising a prediction image of a future frame by means of a second motion compensation from the decoded image of said current frame and the input image of said future frame, wherein:
said first motion compensation and said second motion compensation utilises either positive or negative rounding for pixel value interpolation, and
the rounding methods used in said first and second motion compensations are different.

6. The method of claim 5 wherein information related to the rounding method utilised in said first motion compensation is multiplexed with information related to said difference image and information related to said motion vectors.

7. An image sequence coding method for encoding each frame of an image sequence consisting of a plurality of frames by utilising motion compensation, wherein:

said image sequence contains a plurality of P frames which performs unidirectional prediction in motion com-

pensation, and

P frames using positive and negative rounding for synthesising prediction images appear alternately along the time axis.

8. An image sequence coding method for encoding each frame of an image sequence consisting of a plurality of frames by utilising motion compensation, wherein:

said image sequence includes an I frame and a plurality of P frames, and different rounding methods are used for odd and even
numbered P frames occurring along the time axis from the most recently encoded I frame.

9. An image sequence coding method for encoding each frame of an image sequence consisting of a plurality of frames by utilising motion compensation, wherein a determination whether to use positive or negative rounding for motion compensation of the encoded frame is performed for each frame.

10. The method of claim 9, wherein said determination is made based on a random number generator which outputs numbers corresponding to each of said positive and negative rounding by a possibility close to 50 percent.

11. An image sequence encoder comprising:

a DCT converter for performing DCT conversion to the difference image between the input image of a current frame and the prediction image of said current frame synthesised by motion compensation,
a quantiser for quantising the converted DCT coefficients,
a frame memory for storing the decoded image of a reference frame,
a block matching section for estimating motion vectors and synthesising the prediction image of said current frame by performing motion compensation between the decoded image of said reference frame and the input image of said current frame, and
a multiplexer for multiplexing information related to said quantised DCT coefficients, information related to said motion vectors, and information related to the rounding method used for pixel value interpolation in motion compensation.

12. An image sequence encoder comprising:

a DCT converter for performing DCT conversion to the difference image between the input image of a current frame and the prediction image of said current frame synthesised by motion compensation,
a quantiser for quantising the converted DCT coefficients,
a frame memory for storing the decoded image of a reference frame,
a block matching section for estimating motion vectors and synthesising the prediction image of said current frame by performing motion compensation between the decoded image of said reference frame and the input image of said current frame, and
a multiplexer for multiplexing information related to said quantised DCT coefficients and information related to said motion vectors, wherein:
said block matching section comprises a motion estimator for estimating motion vectors, a prediction image synthesiser for synthesising prediction images, a rounding method determination device for determining whether the rounding method used for motion compensation is positive or negative rounding and outputting the information related to the rounding method determined to use, and
said prediction image synthesiser synthesises said prediction image based on the said information output from said rounding method determination device and information related to said motion vectors output from said motion estimator.

13. The encoder of claim 12, wherein:

said block matching section outputs said information related to the rounding method used for motion compensation to said multiplexer, and
said multiplexer multiplexes said information related to the rounding method with said information related to said quantised DCT coefficients and said information related to said motion vectors.

14. The encoder of claim 12, wherein:

said prediction image synthesiser synthesises prediction images of a plurality of P frames, and different rounding methods are used for odd and even numbered P frames occurring along the time axis from the most recently encoded I frame.

15. An image sequence decoding method comprising the steps of:

extracting information related to motion vectors and information related to quantised DCT coefficients from input information of a decoder,
synthesising a prediction image utilising motion compensation from said motion vectors and the decoded image of a frame decoded in the past, and
synthesising a decoded image by adding said prediction image to an error image obtained by applying dequantisation and inverse DCT conversion to said quantised DCT coefficients,
wherein there are cases when said prediction image is synthesised by motion compensation utilising positive and negative rounding.

16. An image sequence decoding method comprising the steps of:

extracting information related to motion vectors and information related to quantised DCT coefficients from input information of the decoder,
synthesising a prediction image utilizing motion compensation from said motion vectors and a reference image which is the decoded image of a frame previously decoded, and
synthesising a decoded image by adding said prediction image to an error image obtained by applying dequantisation and inverse DCT conversion to said quantised DCT coefficients, wherein:
said motion compensation has half pixel accuracy and the intensity values of chrominance or luminance at a point on the reference image where no pixels are present are calculated by bilinear interpolation, and
there is a case where said bilinear interpolation is calculated by positive rounding according to $Ib = [(La+Lb+1)/2]$, $Ic = [(La+Lc+1)/2]$, and $Id = [(La+Lb+Lc+Ld+2)/4]$, and a case where said bilinear interpolation is calculated by negative rounding according to $Ib = [(La+Lb)/2]$, $Ic = [(La+Lc)/2]$, and $Id = [(La+Lb+Lc+Ld+1)/4]$, where La, Lb, Lc, and Ld are respectively the intensity values of a first pixel, a second pixel which is horizontally adjacent to said first pixel, a third pixel which is vertically adjacent to said first pixel, and a forth pixel which is vertically adjacent to said second pixel and horizontally adjacent to said third pixel, and Ib, Ic, and Id are respectively the interpolated intensity values of the middle point between said first and second pixel, the middle point between said first and third pixel, and the middle point between said first, second, third, and fourth pixel.

17. The method of claim 15 or 16, wherein information specifying either positive or negative rounding is extracted from said input information, and the rounding method specified by said information is used in said motion compensation.

18. An image sequence decoding method for decoding each frame of an image sequence consisting of a plurality of frames, comprising the steps of:

extracting information related to motion vectors and quantised DCT coefficients of a first frame from input information,
synthesising a prediction image of said first frame utilising a first motion compensation from the decoded image of a previously decoded frame and said motion vectors of said first frame,
synthesising a decoded image of said first frame by adding said prediction image of said first frame to an error image obtained by applying dequantisation and inverse DCT to said quantised DCT coefficients of said first frame,
extracting information related to motion vectors and quantised DCT coefficients of a second frame from said input information, and
synthesising a prediction image of a second utilising a second motion compensation from said decoded image of said first frame and said motion vectors of said second frame, wherein:
said first motion compensation and said second motion compensation utilises either positive or negative rounding for pixel value interpolation, and
the rounding methods used in said first motion compensation and said second motion compensation are different.

19. The method of claim 18, wherein information specifying either positive or negative rounding is extracted from said

input information, and the rounding method specified by said information is used in said first motion compensation.

20. An image sequence decoding method for decoding each frame of an image sequence consisting of a plurality of frames by utilising motion compensation, wherein:

said image sequence contains a plurality of P frames which performs unidirectional prediction in motion compensation, and
P frames using positive and negative rounding for synthesising prediction images appear alternately along the time axis.

21. An image sequence decoding method for decoding each frame of an image sequence consisting of a plurality of frames by utilising motion compensation, wherein:

said image sequence includes an I frame and a plurality of P frames, and
different rounding methods are used for odd and even numbered P frames occurring along the time axis from the most recently decoded I frame.

22. An image sequence decoder comprising:

a demultiplexer for extracting information related to motion vectors and information related to quantised DCT coefficients from input information,
a dequantiser for dequantising said quantised DCT coefficients and obtaining DCT coefficients,
an inverse DCT converter to perform inverse DCT conversion to said DCT coefficient and output an error image,
a prediction image synthesiser for synthesising a prediction image by motion compensation using a reference image, which is a previously decoded image, and said motion vectors, and
an adder for adding said error image to said prediction image and output a decoded image,
wherein there are cases where said prediction image synthesiser uses positive and negative rounding for said motion compensation.

23. The device of claim 22 wherein:

said demultiplexer additionally extracts information specifying either positive or negative rounding from said input information, and
said prediction image synthesiser uses the rounding method specified by said information for said motion compensation.

24. An image sequence decoder comprising:

a demultiplexer for extracting information related to motion vectors and information related to quantised DCT coefficients from input information,
a dequantiser for dequantising said quantised DCT coefficients and obtaining DCT coefficients,
an inverse DCT converter to perform inverse DCT conversion to said DCT coefficients and output an error image,
a prediction image synthesiser for synthesising a prediction image by motion compensation using a reference image, which is a previously decoded image, and said motion vectors, and
an adder for adding said error image to said prediction image and output a decoded image,
wherein said prediction image synthesiser has a rounding method determination device to determine whether to use or negative rounding for said motion compensation.

25. The device of claim 24, wherein:

an I frame and a plurality of P frames are decoded, and
different rounding methods are used for odd and even numbered P frames occurring along the time axis from the most recently decoded I frame.

26. A computer readable recording medium on which there is recorded an image sequence encoding method comprising the steps of:

synthesising a prediction image by utilising motion compensation, and

multiplexing information related to the difference image between an input image and said prediction image, and information related to motion vectors estimated during said motion compensation,

wherein information related to the rounding method used for the synthesis of said prediction image is multiplexed with said information related to said difference image and said information related to motion vectors.

27. A computer readable recording medium on which there is recorded an image sequence decoding method comprising the steps of:

extracting information related to motion vectors, information related to quantised DCT coefficients, and information specifying either positive or negative rounding, from input information,

synthesising a prediction image by motion compensation using said motion vectors and the decoded image of a frame decoded in the past, and

a step of synthesising a decoded image by adding said prediction image to an error image obtained by applying dequantisation and inverse DCT conversion to said quantised DCT coefficients,

wherein the rounding method specified by said specifying information is used in said motion compensation.

28. A recording medium on which information related to encoded data of an image sequence is recorded, wherein:

said information related to encoded data of an image sequence is a set of encoded frame information, and said frame information includes information related to the difference image between the input image of the current frame and a prediction image of said current frame synthesised by means of motion compensation, information related to motion vectors estimated by means of said motion compensation, and information for distinguishing whether said motion compensation is motion compensation using positive or negative rounding.

29. An image sequence coding method utilising a motion compensation method assuming non-negative integer values for chrominance and luminance intensity values allowing the vertical and horizontal components of the pixel motion vectors to take values that are not integral multiples of the pixel sampling period in the vertical and horizontal directions, and calculating the intensity value for a point within a reference image where no pixels are present by means of interpolation of sampling values from neighbouring pixels wherein both positive and negative rounding are used in pixel value interpolation in motion compensation, where positive rounding rounds the results of bilinear interpolation with real number operations to the nearest integer and rounds half integer values (0.5 added to an integer) away from zero, and negative rounding differs from positive rounding in that half integer values are rounded towards zero.

30. The method of claim 29, wherein motion compensation using either positive or negative rounding is performed for the luminance signal, and motion compensation using both positive and negative rounding is performed for the chrominance signal.

31. The method of claim 29 or 30 having frames called I frames coded by intraframe coding, P frames coded by interframe coding utilising unidirectional prediction for motion compensation, and B frames coded by interframe coding utilising bi-directional prediction for motion compensation, wherein P frames to which motion compensation using positive and negative rounding are applied to luminance and chrominance signals are located alternately along the time axis.

32. The method of claim 29, wherein:

DCT is used for error coding, and

motion compensation using either positive or negative rounding is performed when the quantisation step size of the DCT coefficients is smaller than a fixed value.

33. The method of claim 29, wherein:

assuming that the pixel sampling period in horizontal and vertical directions are 1, the horizontal and vertical components of a motion vectors are integral multiples of 1/d (d being a positive integer and also an even number), and

motion compensation using either positive or negative rounding is performed when the value of d is larger than a fixed value.

34. The method of claim 29, wherein information indicating whether positive or negative rounding was used for motion compensation for a frame, preferably for the chrominance signal of a frame, is included in the encoded data of said frame.

35. The method of claim 29 having frames called I frames coded by intraframe coding, and P frames coded by interframe coding utilising unidirectional prediction for motion compensation, wherein different rounding methods are used for odd and even numbered P frames occurring along the time axis from the most recently decoded I frame, preferably for the chrominance signal of these P frames.

36. The method of claim 29 which utilises two different motion compensation methods assuming non-negative integer values for chrominance and luminance intensity values, allows the vertical and horizontal components of the pixel motion vectors to take values that are not integral multiples of the pixel sampling period in the vertical and horizontal directions, calculates the intensity value for a point within a reference image where no pixels are present by means of interpolation of sampling values from neighbouring pixels, and obtains a final prediction image by averaging the two prediction images obtained by said two different motion compensation methods, wherein positing averaging is used for said averaging when negative rounding is used for said two different motion compensation methods, and negative averaging is used for said averaging when positive rounding is used for said two different motion compensation methods, where positive averaging rounds the results of averaging with real number operations to the nearest integer and rounds half integer values (0.5 added to an integer) away from zero, and negative averaging differs from positive averaging in that half integer values are rounded towards zero.

37. An image sequence coding method which utilises two different motion compensation methods assuming non-negative integer values for chrominance and luminance intensity values, and obtains a final prediction image by averaging the two prediction images obtained by said two different motion compensation methods, wherein both positing and negative averaging is used for said averaging, where positive averaging rounds the results of averaging with real number operations to the nearest integer and rounds half integer values (0.5 added to an integer) away from zero, and negative averaging differs from positive averaging in that half integer values are rounded towards zero.

38. A method of coding images comprising the steps of:

storing a reference image,
synthesising a prediction image by performing motion compensation between said reference image and an input image, and
generating multiplexed information including information of said motion vectors and information specifying a rounding method which is used for pixel value interpolation in said motion compensation.

39. A method of coding images comprising the steps of:

storing a reference image, and
performing motion compensation by comparing an input image and said reference image to estimate motion vectors and synthesise a prediction image,
wherein a rounding method which is used for pixel value interpolation in said motion compensation for synthesising said prediction image is different from a rounding method used for synthesising a prediction image of said reference image.

40. The method of claim 39, further comprising generating multiplexed information including information of said motion vectors and information specifying said rounding method used for synthesising said prediction image.

41. An image encoder comprising:

a memory for storing a reference image,
a block matching section to synthesise a prediction image by performing motion compensation between said reference image and an input image, and estimate motion vectors, and
a multiplexer to generate multiplexed information including information of said motion vectors and information

specifying a rounding method which is used for pixel value interpolation in said motion compensation.

42. An image encoder comprising:

a memory for storing a reference image which is a previously decoded image,
a block matching section to synthesise a prediction image by performing motion compensation between an input image and said reference image and estimate motion vectors, and
a rounding method determination device for determining the rounding method used for pixel value interpolation in said motion compensation.

43. A method of decoding images comprising the steps of:

storing a reference image which is a previously decoded image,
receiving information of motion vectors, and
performing motion compensation to synthesise a prediction image by using said motion vectors and said reference image,
wherein the rounding method which is used for pixel value interpolation in said motion compensation for synthesising said prediction image is different from a rounding method used for synthesising a prediction image of said reference image.

44. A method of decoding image comprising the steps of:

storing a reference image,
receiving information including information of motion vectors and information specifying a rounding method which is used for synthesising a prediction image in an encoder, and
synthesising a prediction image by performing motion compensation using said motion vectors and said reference image,
wherein the rounding method used for pixel value interpolation in said motion compensation is controlled according to said information specifying a rounding method.

45. An image decoder comprising:

a memory for storing a reference image, which is a previously decoded image, and
a synthesiser for synthesising a prediction image by performing motion compensation using received motion vectors and said reference image,
wherein said synthesiser controls the rounding method used for pixel value interpolation in said motion compensation, so that the rounding method used for synthesising said prediction image is different from that used for synthesising a prediction image of said reference image.

46. An image decoder comprising:

a memory for storing a reference image, which is a previously decoded image, and
a synthesiser for synthesising a prediction image by performing motion compensation using received motion vectors and said reference image,
wherein said synthesiser receives information specifying the rounding method used in encoding process and controls the rounding method used for pixel value interpolation in said motion compensation according to said information specifying the rounding method.

47. The invention of any of claims 38 to 46, wherein said rounding is positive or negative.

48. A recording medium for recording information of images which is encoded by performing motion compensation, wherein said information includes information of specifying a rounding method used for pixel value interpolation in said motion compensation.

49. The recording medium of claim 48, wherein said information specifying a rounding consists of one bit.

FIG. 1

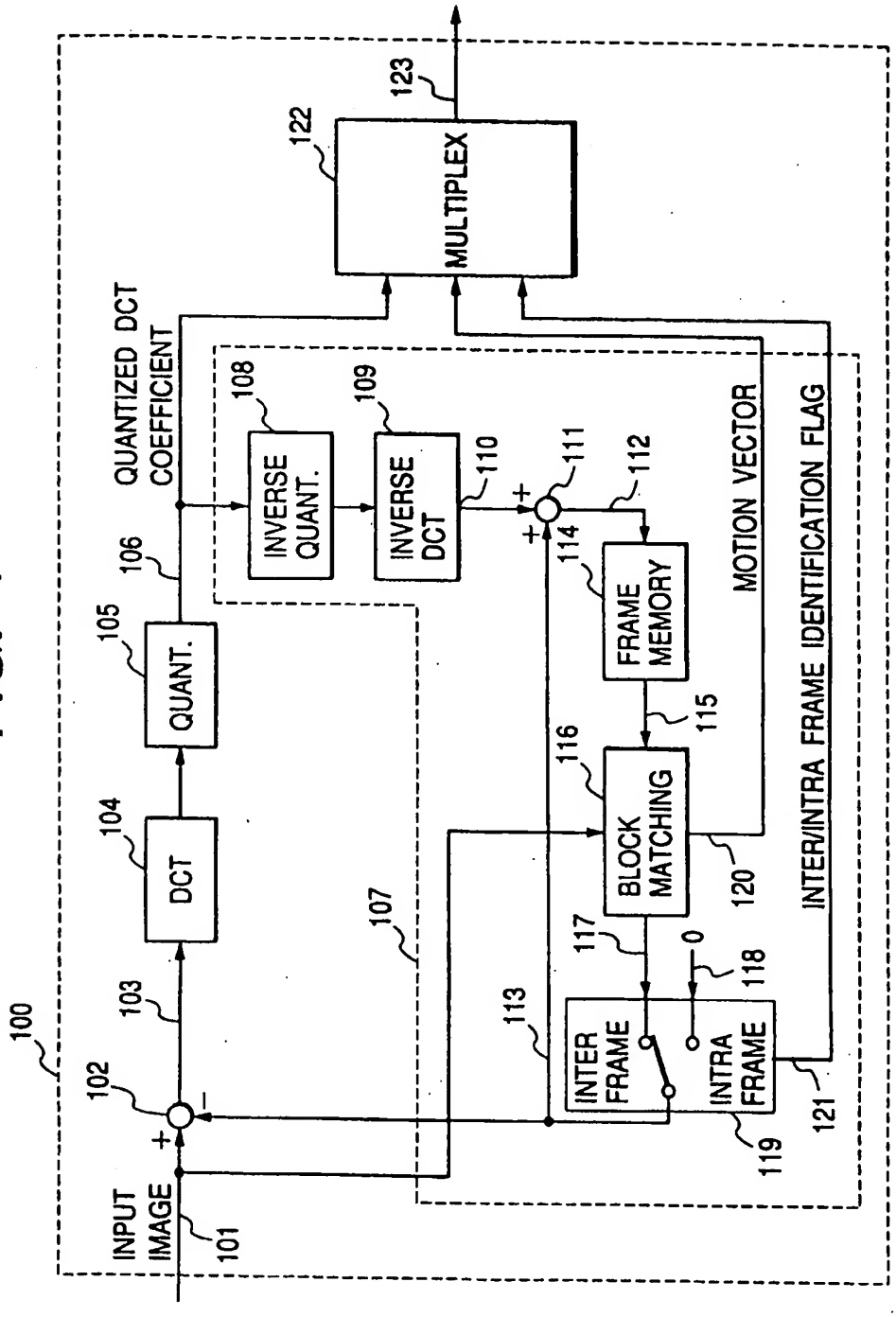

100
INPUT IMAGE
101
102
103
DCT
104
QUANT.
105
106
QUANTIZED DCT COEFFICIENT
107
INVERSE QUANT.
108
INVERSE DCT
109
110
111
112
FRAME MEMORY
114
115
BLOCK MATCHING
116
117
INTER FRAME
113
INTRA FRAME
118
119
0
120
MOTION VECTOR
121
INTER/INTRA FRAME IDENTIFICATION FLAG
122
MULTIPLEX
123

FIG. 2

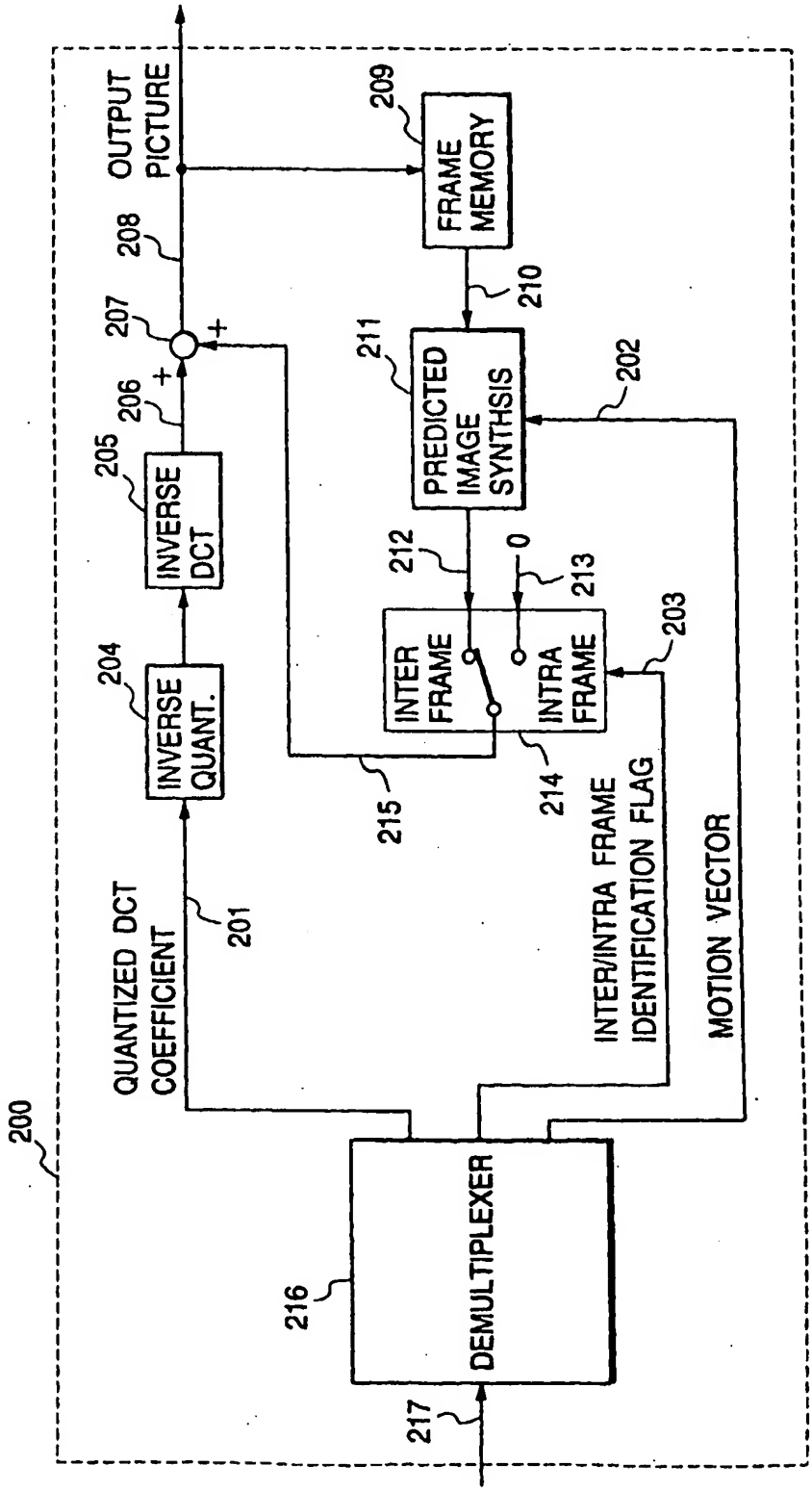

200
OUTPUT PICTURE
208
209
FRAME MEMORY
210
207
+
206
+
211
PREDICTED IMAGE SYNTHSIS
202
205
INVERSE DCT
212
0
213
204
INVERSE QUANT.
INTER FRAME
INTRA FRAME
203
215
214
QUANTIZED DCT COEFFICIENT
201
INTER/INTRA FRAME IDENTIFICATION FLAG
MOTION VECTOR
216
DEMULTIPLEXER
217

FIG. 3

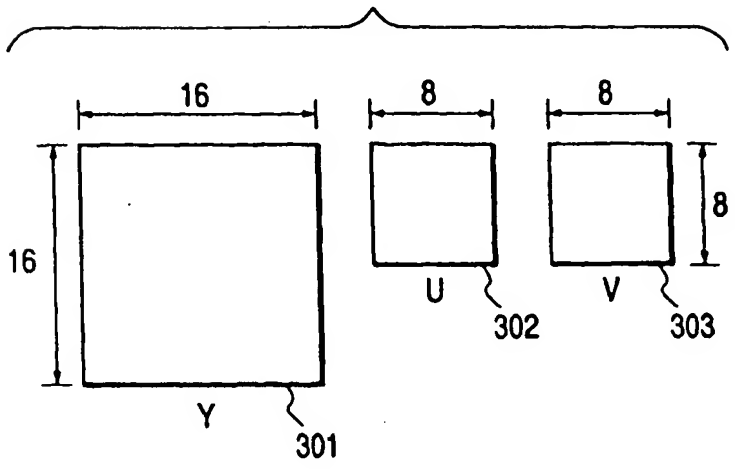

16
8
8
16
8
U
V
302
303
Y
301

FIG. 4

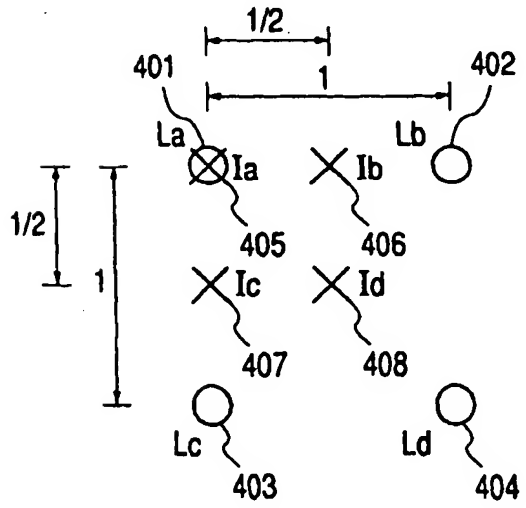

1/2
401
1
402
La
Lb
Ia
Ib
1/2
405
406
1
Ic
Id
407
408
Lc
Ld
403
404

FIG. 5

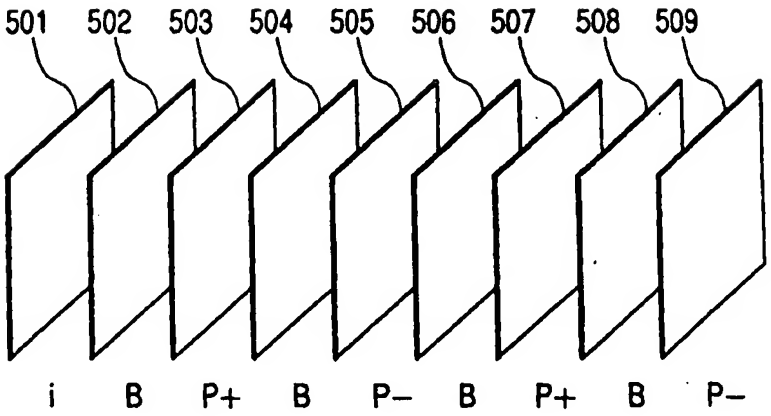


FIG. 6

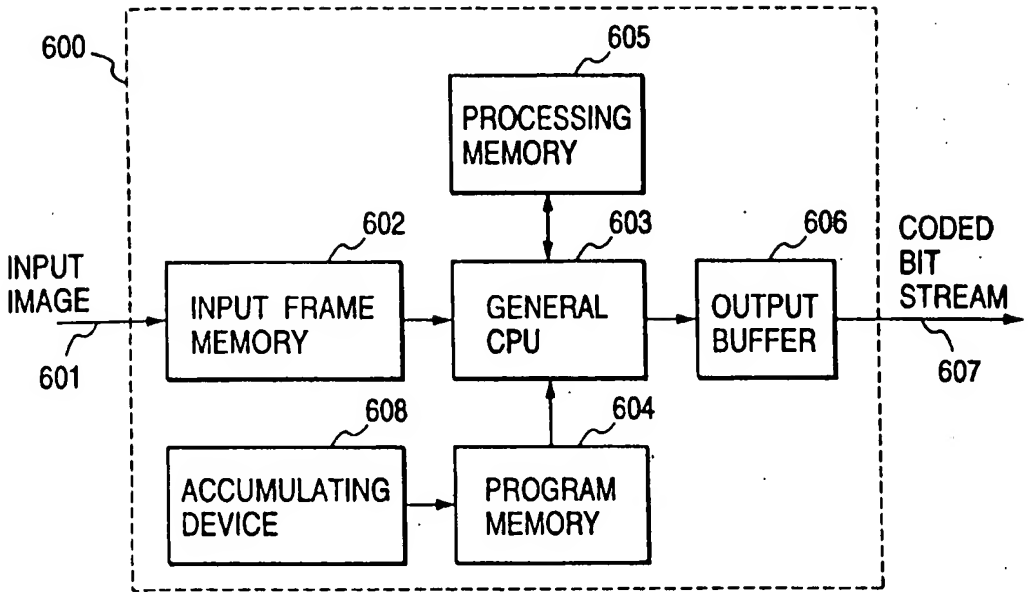

600
605
PROCESSING MEMORY
602
603
606
CODED BIT STREAM
INPUT IMAGE
INPUT FRAME MEMORY
GENERAL CPU
OUTPUT BUFFER
601
607
608
604
ACCUMULATING DEVICE
PROGRAM MEMORY

FIG. 7

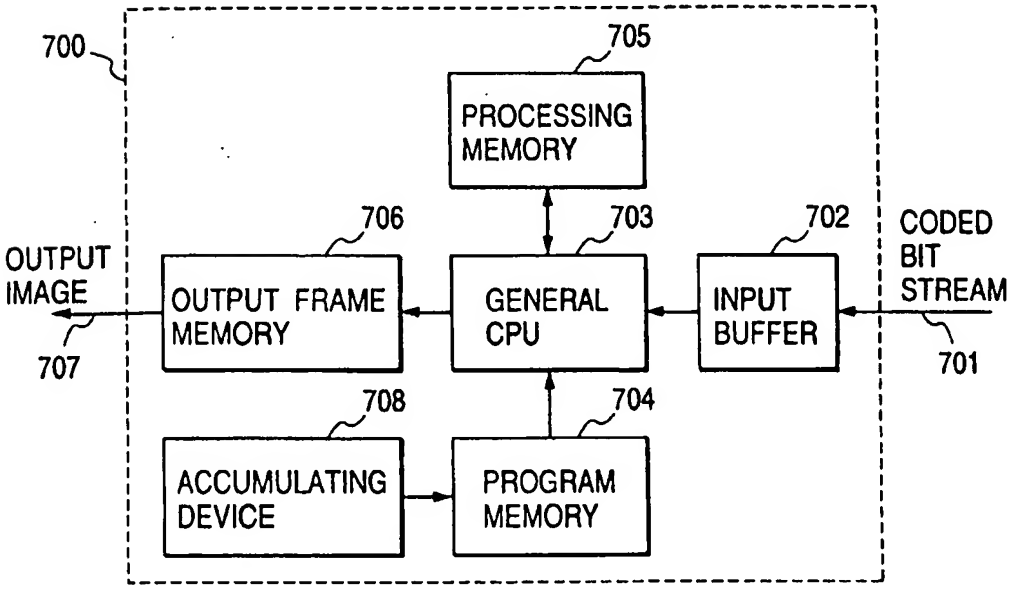

700
705
PROCESSING MEMORY
706
703
702
CODED BIT STREAM
OUTPUT IMAGE
OUTPUT FRAME MEMORY
GENERAL CPU
INPUT BUFFER
707
701
708
704
ACCUMULATING DEVICE
PROGRAM MEMORY

FIG. 8

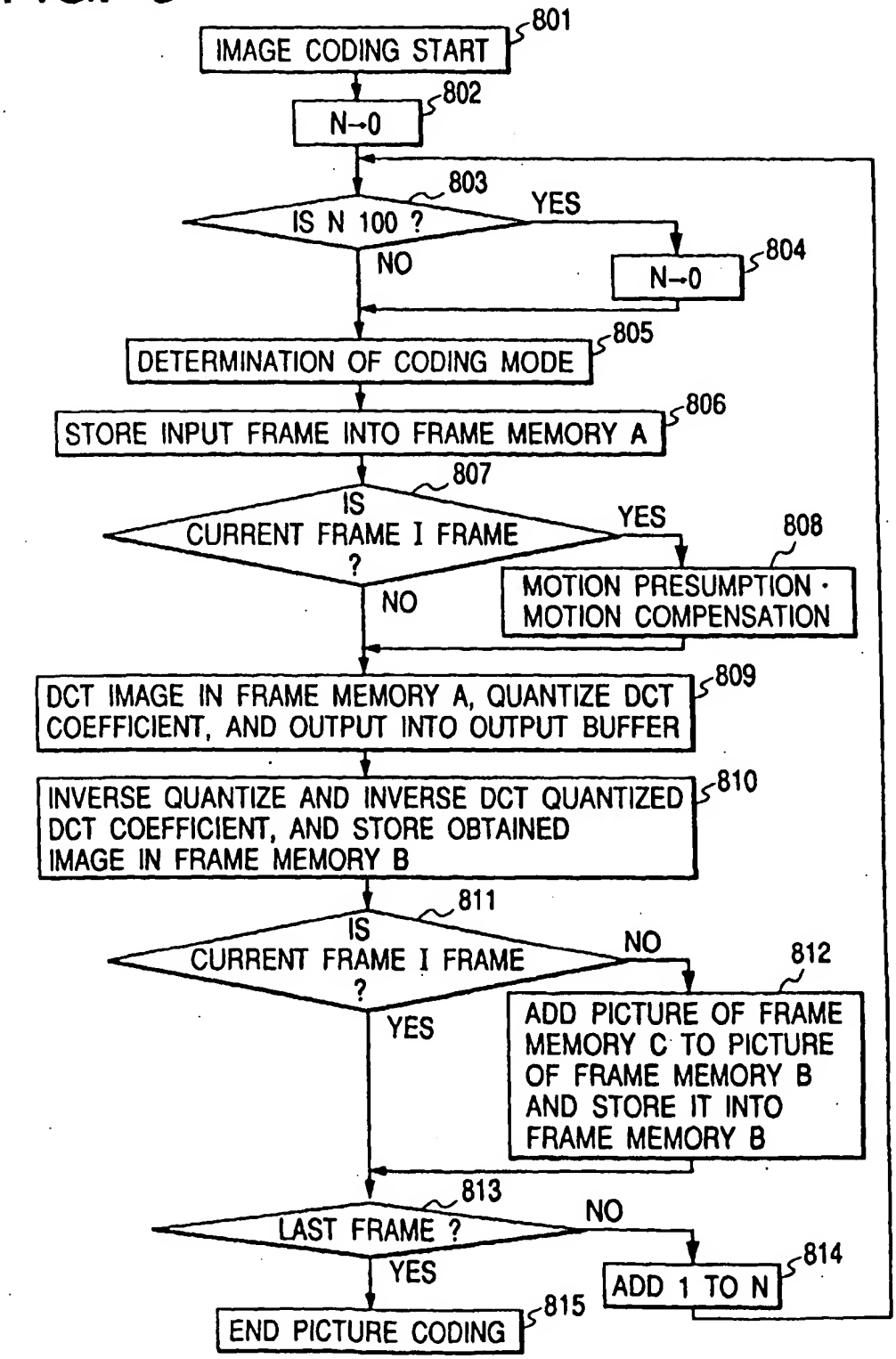


IMAGE CODING START
801
N→0
802
IS N 100 ?
803
YES
NO
N→0
804
DETERMINATION OF CODING MODE
805
STORE INPUT FRAME INTO FRAME MEMORY A
806
IS CURRENT FRAME I FRAME ?
807
YES
NO
MOTION PRESUMPTION · MOTION COMPENSATION
808
DCT IMAGE IN FRAME MEMORY A, QUANTIZE DCT COEFFICIENT, AND OUTPUT INTO OUTPUT BUFFER
809
INVERSE QUANTIZE AND INVERSE DCT QUANTIZED DCT COEFFICIENT, AND STORE OBTAINED IMAGE IN FRAME MEMORY B
810
IS CURRENT FRAME I FRAME ?
811
NO
YES
ADD PICTURE OF FRAME MEMORY C TO PICTURE OF FRAME MEMORY B AND STORE IT INTO FRAME MEMORY B
812
LAST FRAME ?
813
NO
YES
ADD 1 TO N
814
END PICTURE CODING
815

FIG. 9

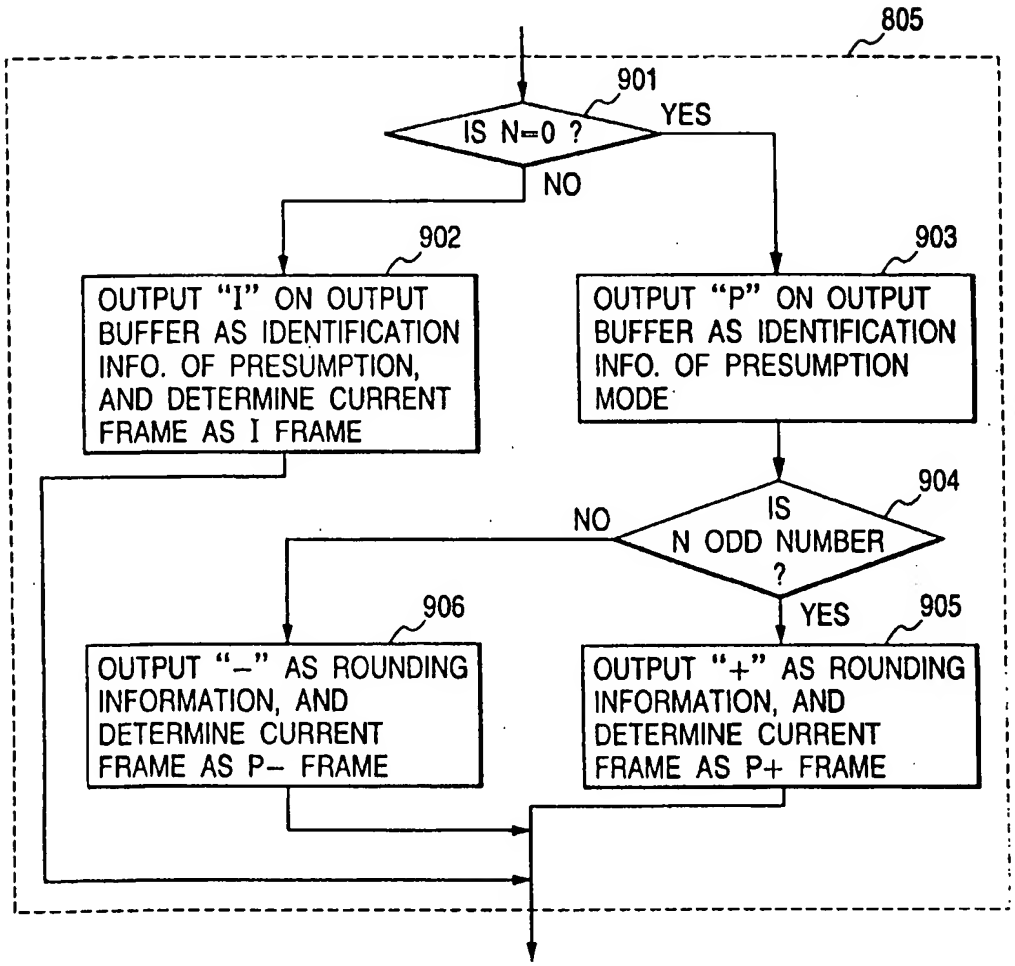

805
901
IS N=0 ?
YES
NO
902
OUTPUT "I" ON OUTPUT BUFFER AS IDENTIFICATION INFO. OF PRESUMPTION, AND DETERMINE CURRENT FRAME AS I FRAME
903
OUTPUT "P" ON OUTPUT BUFFER AS IDENTIFICATION INFO. OF PRESUMPTION MODE
904
IS N ODD NUMBER ?
NO
YES
906
OUTPUT "−" AS ROUNDING INFORMATION, AND DETERMINE CURRENT FRAME AS P− FRAME
905
OUTPUT "+" AS ROUNDING INFORMATION, AND DETERMINE CURRENT FRAME AS P+ FRAME

FIG. 10

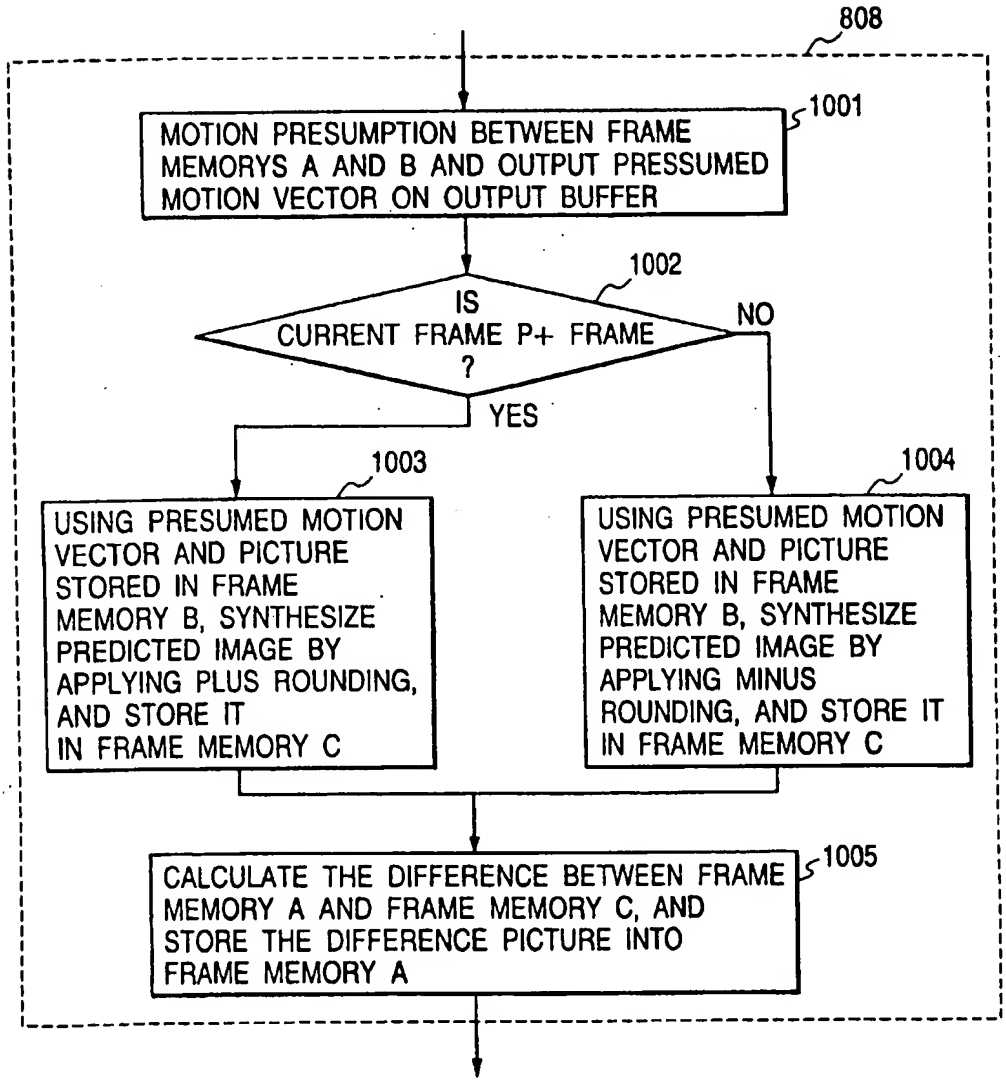

808
1001
MOTION PRESUMPTION BETWEEN FRAME MEMORYS A AND B AND OUTPUT PRESSUMED MOTION VECTOR ON OUTPUT BUFFER
1002
IS CURRENT FRAME P+ FRAME ?
NO
YES
1003
USING PRESUMED MOTION VECTOR AND PICTURE STORED IN FRAME MEMORY B, SYNTHESIZE PREDICTED IMAGE BY APPLYING PLUS ROUNDING, AND STORE IT IN FRAME MEMORY C
1004
USING PRESUMED MOTION VECTOR AND PICTURE STORED IN FRAME MEMORY B, SYNTHESIZE PREDICTED IMAGE BY APPLYING MINUS ROUNDING, AND STORE IT IN FRAME MEMORY C
1005
CALCULATE THE DIFFERENCE BETWEEN FRAME MEMORY A AND FRAME MEMORY C, AND STORE THE DIFFERENCE PICTURE INTO FRAME MEMORY A

FIG. 11

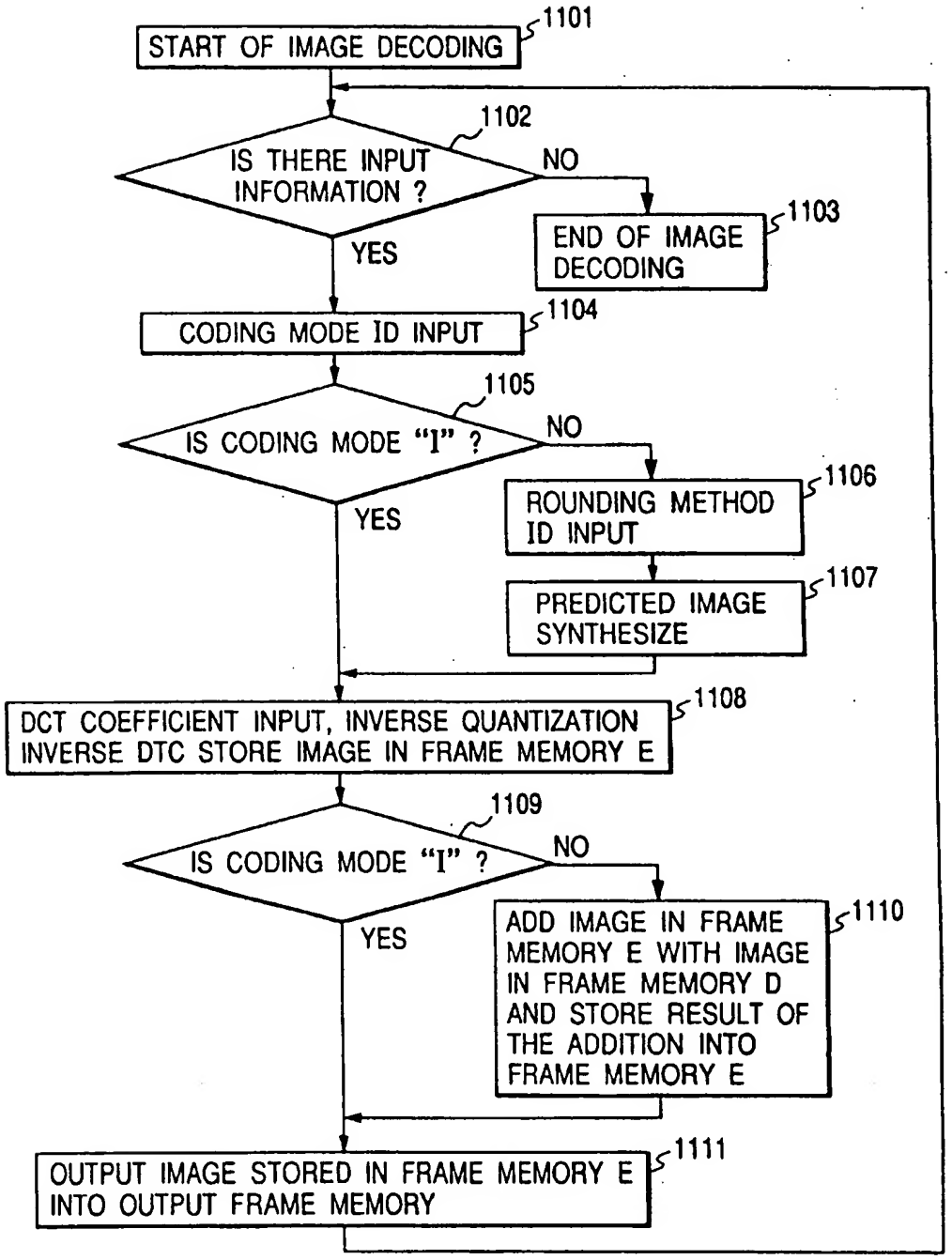

START OF IMAGE DECODING
1101
IS THERE INPUT INFORMATION ?
1102
NO
YES
END OF IMAGE DECODING
1103
CODING MODE ID INPUT
1104
IS CODING MODE "I" ?
1105
NO
YES
ROUNDING METHOD ID INPUT
1106
PREDICTED IMAGE SYNTHESIZE
1107
DCT COEFFICIENT INPUT, INVERSE QUANTIZATION INVERSE DTC STORE IMAGE IN FRAME MEMORY E
1108
IS CODING MODE "I" ?
1109
NO
YES
ADD IMAGE IN FRAME MEMORY E WITH IMAGE IN FRAME MEMORY D AND STORE RESULT OF THE ADDITION INTO FRAME MEMORY E
1110
OUTPUT IMAGE STORED IN FRAME MEMORY E INTO OUTPUT FRAME MEMORY
1111

FIG. 12

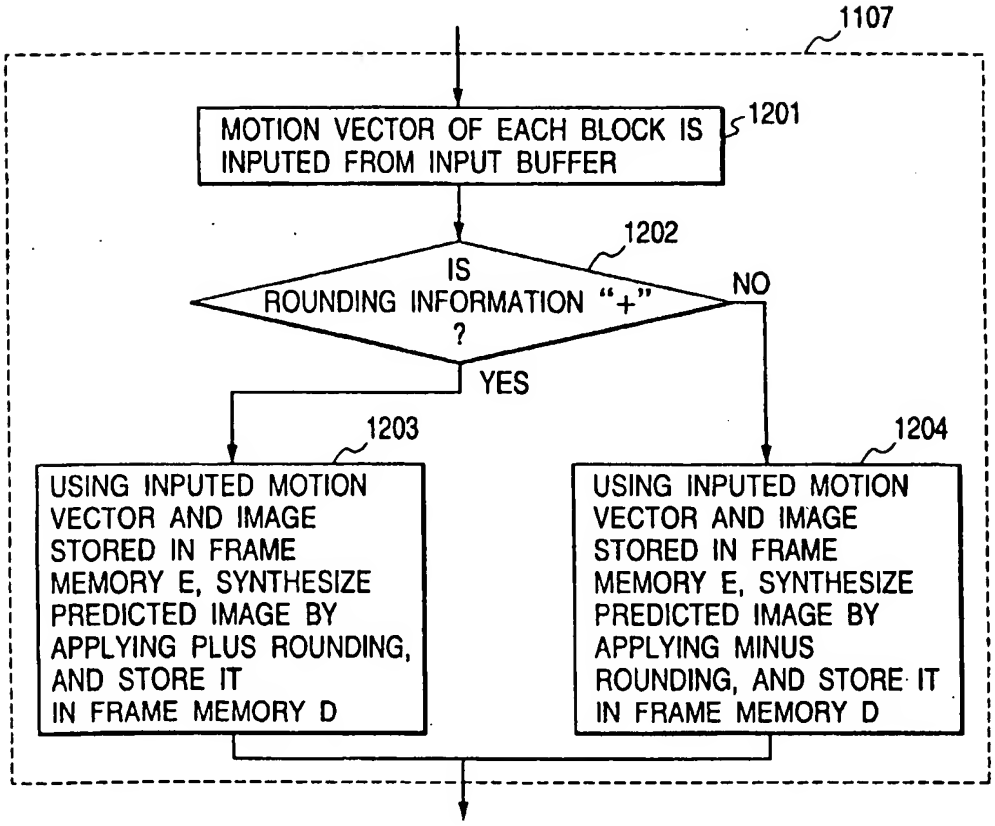

1107
MOTION VECTOR OF EACH BLOCK IS INPUTED FROM INPUT BUFFER
1201
1202
IS ROUNDING INFORMATION "+" ?
NO
YES
1203
USING INPUTED MOTION VECTOR AND IMAGE STORED IN FRAME MEMORY E, SYNTHESIZE PREDICTED IMAGE BY APPLYING PLUS ROUNDING, AND STORE IT IN FRAME MEMORY D
1204
USING INPUTED MOTION VECTOR AND IMAGE STORED IN FRAME MEMORY E, SYNTHESIZE PREDICTED IMAGE BY APPLYING MINUS ROUNDING, AND STORE IT IN FRAME MEMORY D

FIG. 13

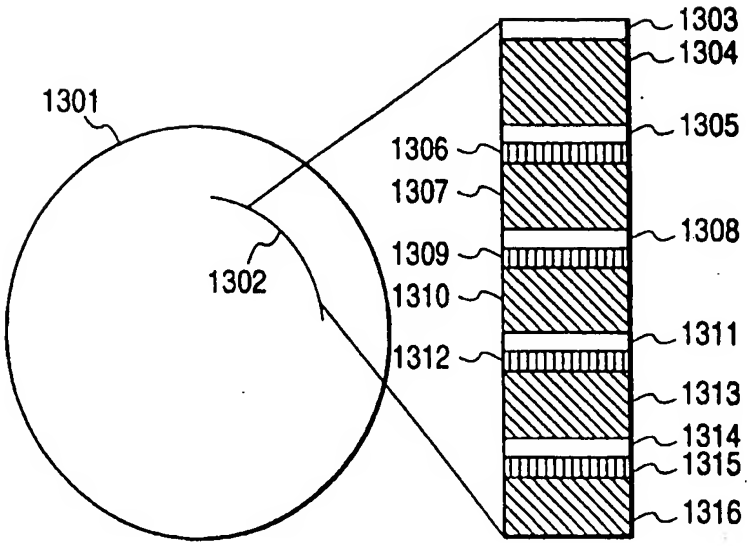

1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316

FIG. 15

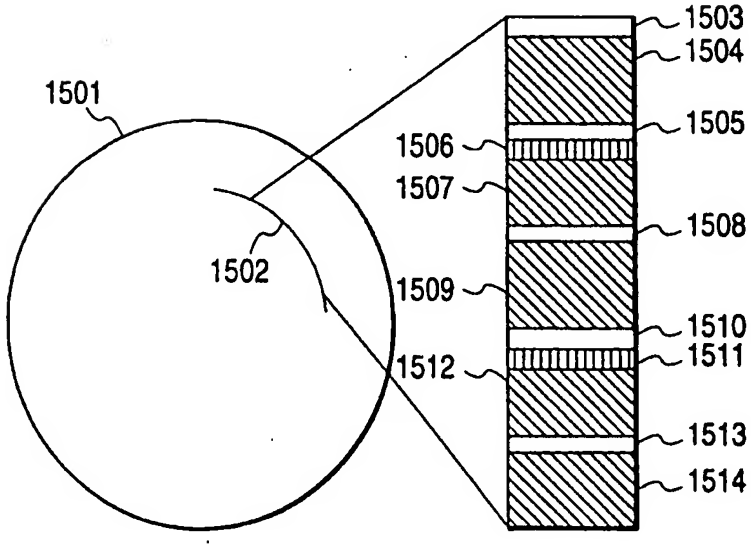

1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514

FIG. 14

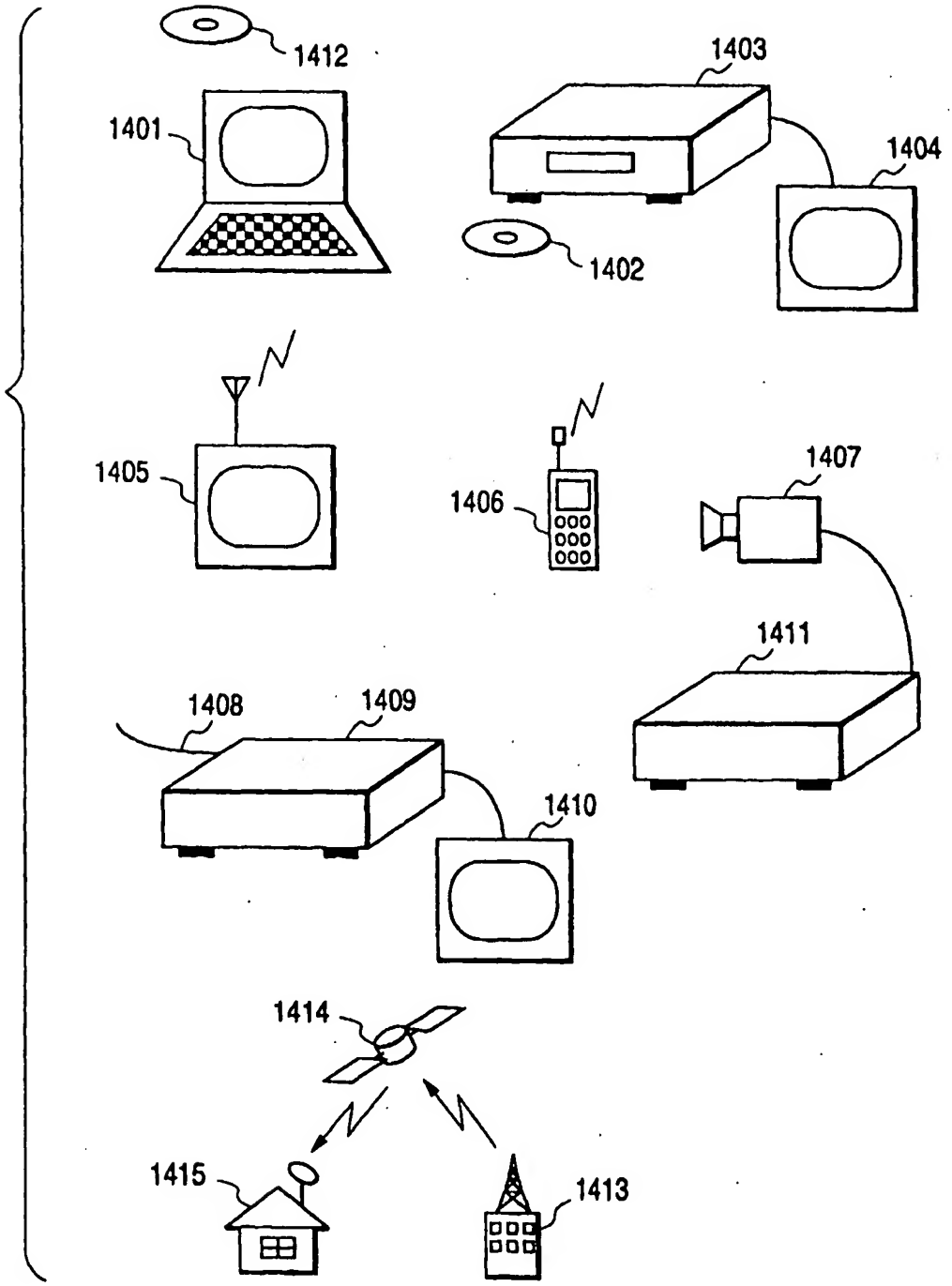

1412
1401
1403
1404
1402
1405
1406
1407
1411
1408
1409
1410
1414
1415
1413

FIG. 16

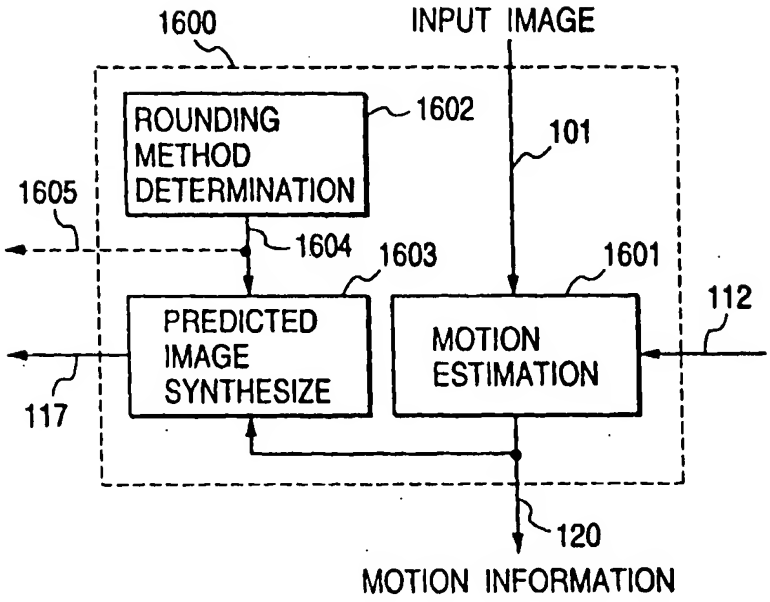
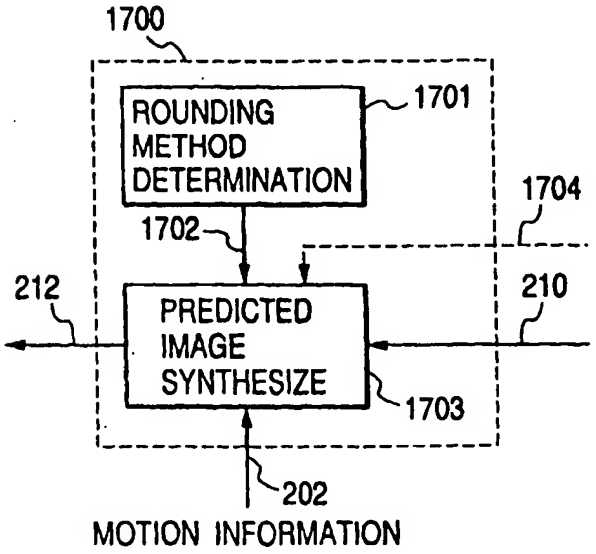

1600
INPUT IMAGE
ROUNDING METHOD DETERMINATION
1602
101
1605
1604
1603
1601
112
PREDICTED IMAGE SYNTHESIZE
MOTION ESTIMATION
117
120
MOTION INFORMATION

FIG. 17


1700
ROUNDING METHOD DETERMINATION
1701
1704
1702
212
210
PREDICTED IMAGE SYNTHESIZE
1703
202
MOTION INFORMATION